SONDERHEFT 13 ÖS 100,-/sfr 14,- Lit 12.000/hfl 18,-/dkr 118,- DM 14,-

6. SCHNEIDER SONDERHEFT

# HAPPY-COMPUTER

Markt&Technik

DAS GROSSE HEIMCOMPUTER-MAGAZIN

## Schneider CPC: Die besten Textprogramme im Vergleich

## CPC-Pannenhilfe!

Kleine Fehler schnell behoben

## Disketten-Wissen

- ★ So liest man Fremdformate
- ★ Super Diskettendoktor selbst programmiert

## MS-DOS: Die Alternative

Nutzen, Kosten, Einstieg

## Spitzen-Listings

- ★ Anwendungen
- ★ Tips&Tricks
- ★ CP/M

Alle Programme auf Diskette und Kassette erhältlich



ocean
DIGITAL INTEGRATION
ACTIVISION
HOME COMPUTER SOFTWARE
They sold a
MILLION
#3
FIGHTER PILOT
Vier Super-Spiele zum Preis von einem: Eine Flugsimulation, ein Karate-Knüller, Weltraum-Action und ein Kino-Hit!
Daley Thompson's SUPER-TEST
PISTOL SHOOTING
TUG O' WAR
SPRING BOARD DIVING
CYCLING
GIANT SLALOM
PENALTIES
KUNG-FU MASTER
THE OFFICIAL COIN-OP GAME
GHOSTBUSTERS
Auf Kassette und Diskette für Commodore 64/128 Schneider CPC Spectrum 48K mit deutscher Anleitung
ACTIVISION.
Ghostbusters™ is a trademark of Columbia Pictures Industries, Inc.
Vorsicht vor Grauimporten!
Bitte prüfen Sie schon beim Kauf, ob dieses Programm wirklich eine deutsche Anleitung enthält. Spätere Reklamationen können leider nicht berücksichtigt werden.
The Hit SQUAD
Ocean Deutschland, An der Gümpgesbrücke 24, 4044 Kaarst 2
Vertrieb: Rushware · Mitvertrieb: MICRO-HÄNDLER · Distribution in Österreich: Karasoft
Ocean-Produkte erhalten Sie in den Fachabteilungen von KAUFHOF und Quelle sowie in allen gutsortierten Computershops und im guten Versandhandel



# Ein Viertel-jahr danach

Drei Monate ist es her, seit Schneider den Einstieg in die PC-Klasse wagte. Wie man heute sieht – mit Erfolg. Die etwas mehr als 30000 Computer, die 1986 noch verkauft werden sollten, haben in nur wenigen Wochen ihren Weg zum Käufer gefunden. Wer in der letzten Zeit noch ein Gerät haben wollte, mußte sich damit begnügen, auf die Warteliste gesetzt zu werden. Erinnern Sie sich, wie 1984 der CPC 464 auf dem deutschen Markt eingeführt wurde?

Aber die Wogen des ersten Ansturms auf den PC glätten sich und der Schneider CPC tritt wieder aus dem Schatten seines großen Bruders. Mit einem Preis von knapp 1000 Mark ist der CPC 6128 für den Hobbyisten sicher das geeignetere Gerät. Und was man mit so einem »Kleinen« alles machen kann! Auf den 164 Seiten des inzwischen 6. Schneider-Sonderhefts von Happy-Computer gibt es wieder für jeden etwas. Wir haben uns die Umfragen aus den letzten beiden Schneider-Sonderheften zu Herzen genommen und Ihre Anregungen gezielt berücksichtigt.

Fast jeder unserer Leser will noch mehr Tips&Tricks zu seinem Computer. Wir haben also wieder alle gefragt, die sich mit Schneider, mit CP/M oder mit MS-DOS auskennen, ob sie wieder Wissenswertes weitergeben wollen. Und wie sie wollten!

Wer seinen Computer nicht nur für das Hobby, sondern auch professionell benutzen will, der findet einen Terminkalender, eine universelle Dateiverwaltung und für Mathe-Fans eine komplette Fourier-Analyse. Spiele und Utilities runden das Angebot an Listings ab.

Falls das Thema eines Programms Sie nicht interessiert, schauen Sie es sich trotzdem einmal an. Alle Programme überzeugen nicht nur durch ihre Leistungsfähigkeit, sondern auch durch ihren trickreichen Aufbau. So lernen selbst absolut Mathe-Unlustige bei der Fourier-Analyse einiges dazu.

Jeder zweite unserer Leser will mehr Informationen über neue Produkte. Eigentlich unverständlich, da im Stammheft der Happy-Computer alles vorgestellt wird, was es Interessantes für Schneider gibt. Nichtsdestotrotz haben wir die wichtigsten Neuerscheinungen der letzten drei Monate zusammengefaßt. Ein Blick auf diese Produktvorstellungen sagt Ihnen mehr als die sonst üblichen Kurzbeschreibungen.

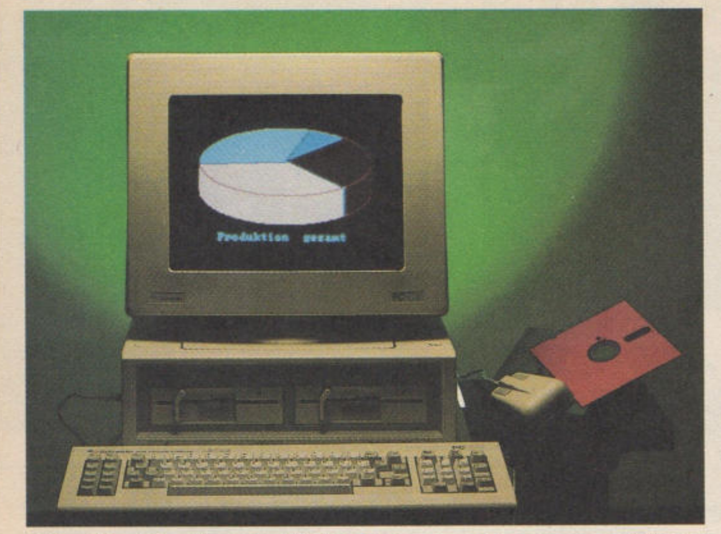
Eigentlich wollten wir Ihnen in diesem Heft auch ausführliche Informationen über die MS-DOS-Karten für die Schneider CPC geben. Doch leider blieben die Bemühungen der Entwicklungsingenieure hinter den Wünschen der Verkaufschefs zurück. Die Frage vieler Schneider-Fans, ob sie jetzt umsteigen sollen, sich einen Zweit-Computer oder den schon vorhandenen aufrüsten sollen, bleibt weiterhin offen. Der erste Emulator ist inzwischen ausverkauft. Der Hersteller in Aachen arbeitet an einer Neuauflage der Serie. Und auch die württembergische Konkurrenz ist noch am »basteln«. Dieses Thema müssen wir also zwangsläufig für eine der nächsten Ausgaben von Happy-Computer aufsparen.

»In der Happy-Computer steht ja nur sehr wenig über den Schneider CPC!«, lautet eine andere, oft gehörte Kritik. Aber stimmt das wirklich? Mehr als 20 Seiten Spieletests, jede Menge CP/M-Tips, die Wordstar-Werkstatt (um nur eins der vielen Software-Themen anzusprechen) oder alles über Drucker, DFÜ und so weiter – das sind alles Themen der Happy-Computer, die ausführliche Informationen bieten.

Nun aber viel Spaß mit dem neuesten Schneider-Sonderheft. Sagen Sie uns, was wir besser – oder weiter so – machen sollen. Jeder Ihrer Briefe wird beantwortet. Denn nur mit Ihrer Hilfe können wir über das berichten, was Sie lesen wollen. Ihre Meinung heute ist unsere Grundlage für das Sonderheft von morgen. Den aktiv mitwirkenden Lesern am nächsten Sonderheft sagen wir heute schon Danke.

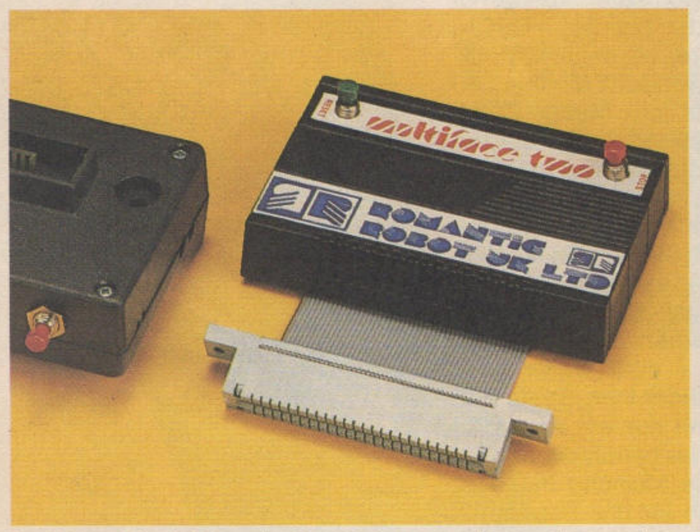
Ihr
Andreas Hagedorn

# Inhalt



Schneider ist voll auf den MS-DOS-Geschmack gekommen. Ob der IBM-kompatible PC oder ein MS-DOS-Emulator für Ihren CPC: Wer den Einstieg wagen will, ist bei Schneider immer gut aufgehoben.
33

Ob Sie nun ein Spielefreak sind oder nicht: Es gibt gewisse Spiele, die in einer Softwaresammlung nicht fehlen dürfen. Wir stellen Ihnen diese Juwelen der Schneider-Spiele-Szene mit vielen bunten Bildern vor.
16




Kopiermodule heißen die Tausendsassas, die vor kaum einem Kopierschutz die Waffen strecken. Zwei leistungsfähige Konkurrenten stellten sich einem ausführlichen Test und – die Auswahl fällt nicht leicht.
12

## Aktuell



## Hardware



## Spiele



## Bastelei



## MS-DOS: Die Alternative



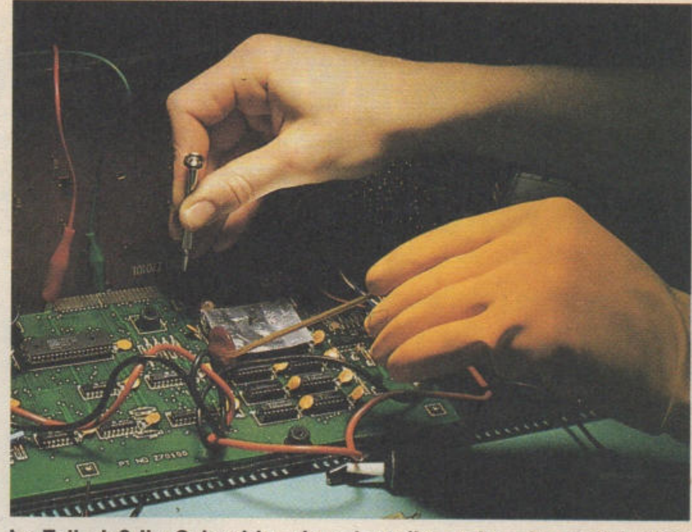
## Einsteiger



## Software



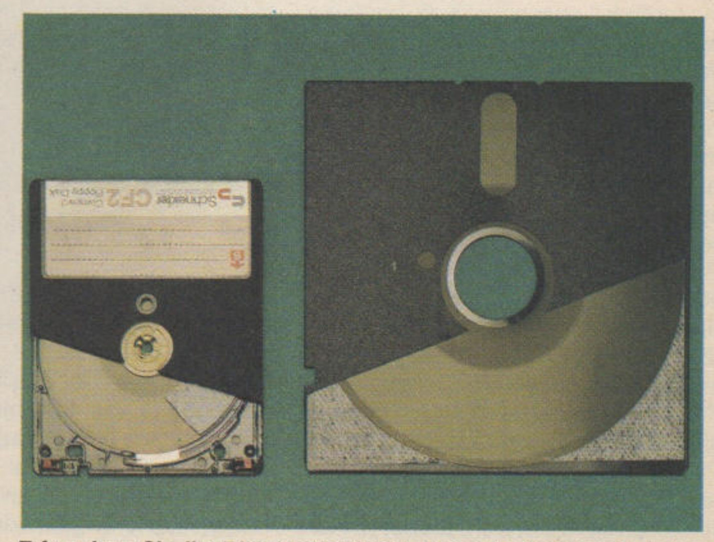
## Listings



Im Fall, daß Ihr Schneider einmal streikt, stehen Sie meist ratlos vis-à-vis. Doch das Gerät gleich zur Reparatur zu geben ist oft nicht nötig. Wir geben Ihnen hilfreiche Tips, wie Sie Ihrem CPC wieder auf die Sprünge helfen. 18



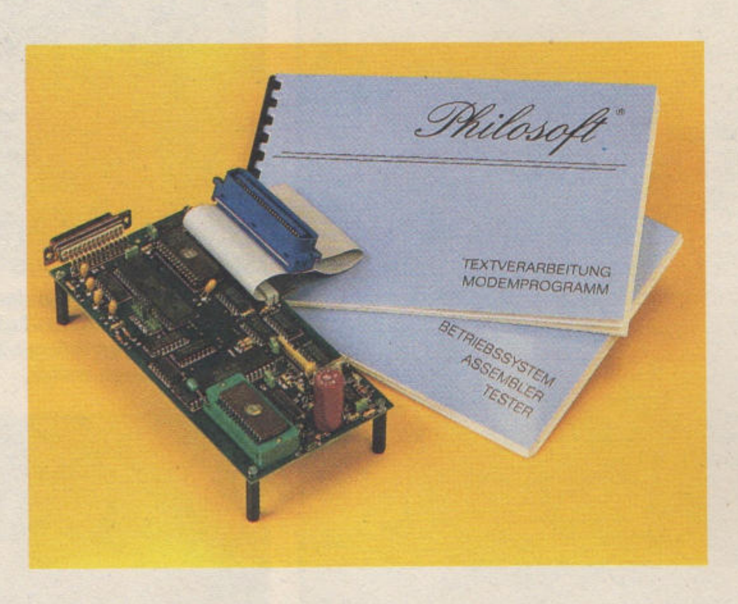
Wir vermitteln Ihnen mit Ihrem Schneider den Einstieg in die Datenfernübertragung. Was Sie im einzelnen dazu brauchen und welche ungeahnten Möglichkeiten sich mit diesem System eröffnen, sagt Ihnen unser Beitrag. 48




Erforschen Sie Ihr Diskettenlaufwerk. Unser Schwerpunkt informiert über Controller, Laufwerk-Routinen, Diskettenformate und vieles Interessante mehr. Sogar einen super Diskettendoktor können Sie mit diesem Wissen selbst programmieren. 129

# Multitalent für Schneider CPC

**Eine universelle Hardware-Erweiterung, passend für alle CPC-Modelle, hat Philosoft entwickelt. Die höchste Ausbaustufe enthält 32 KByte leistungsfähige Software, eine serielle Schnittstelle und ein EPROM-Programmiergerät.**




**Die vollständig ausgebaute Erweiterung mit Handbüchern**

CPC-Besitzer, die Ihren Computer ausbauen möchten, kennen das Problem: Jede Funktion (Schnittstelle, Zusatzspeicher etc.) erfordert seine eigene Erweiterung. Beim Anschluß mehrerer Erweiterungen an den CPC ist der Kabelsalat perfekt. Die ausbaufähige Hardware-Erweiterung von Philsosoft schafft Abhilfe. Sie vereinigt mehrere Funktionen, die ein engagierter Anwender benötigt, auf einer einzigen Platine.

Die Grundausstattung zum Preis von 270 Mark besteht aus einer teilweise bestückten Platine, die auf vier Beinchen steht und über ein Flachbandkabel mit dem Erweiterungsanschluß des CPC verbunden wird. Alle Leitungen des Erweiterungsanschlusses sind durchgeschleift, so daß zusätzliche Peripherie (Laufwerk, Digitalisierer, etc.) an die Platine angeschlossen werden kann. Zur Dokumentation sind zwei Handbücher beigelegt.

Positiv fällt auf, daß für den Schaltungsaufbau ausschließlich hochwertige Bauelemente verwendet wurden.

In der Grundausstattung ist die Platine mit einem 32 KByte EPROM, der ein eigenes Betriebssystem und mehrere Anwendungsprogramme enthält, vier ICs und einigen anderen Kleinteilen bestückt. Für die Nachrüstung einer seriellen Schnittstelle und eines EPROM-Programmiergeräts sind bereits alle Leiterbahnen, Anschlüsse und Bohrungen vorhanden.

Das EPROM arbeitet als Erweiterungs-ROM für das Betriebssystem des CPC. Die vier ICs bilden eine Decodierlogik, die den Zugriff des CPC auf das EPROM regelt. Nach dem Einschalten des CPC führt das Betriebssystem eine Initialisierung durch. In dieser Phase wird der Erweiterungsanschluß nach zusätzlichen ROMs abgefragt. Das EPROM wird über die Decodierlogik erkannt und in das Betriebssystem eingebunden.

Das Betriebssystem unterscheidet die Erweiterungs-ROMs durch eine Erkennungsnummer. Wenn neben dem EPROM noch weitere ROMs angeschlossen sind (zum Beispiel das Disketten-ROM) und sich die Nummern überschneiden, kann man die Nummer des EPROMs über einen Jumper (steckbare Drahtbrücke) verändern.

Der Aufruf des EPROMs aus dem Basic des CPC heraus erfolgt über einen kurzen RSX-Befehl. Hierdurch gelangt man in eine CP/M-ähnliche Programmierebene, wo dem Programmierer 24 teilweise Basic-kompatible Befehle zur Verfügung stehen. Beim Anschluß eines Diskettenlaufwerks können zusätzlich 13 Befehle des Amsdos genutzt werden.

## Alles was das Herz begehrt

Neben den Systembefehlen enthält das EPROM vier leistungsfähige Anwendungsprogramme. Die Programme rufen Sie aus der Systemebene durch Eingabe eines Befehls auf. Der Programmstart erfolgt augenblicklich, da lediglich ein Sprung an eine bestimmte EPROM-Adresse notwendig ist. Durch eine Aktivierungsmeldung und Ausgabe eines Prompts zeigt jedes Programm seine Betriebsbereitschaft an.

Mit dem Befehl EDIT starten Sie ein komfortables Textverarbeitungsprogramm (96 Kommandos), das zu »Wordstar« weitgehend kompatibel ist. Auch das Format, das Wordstar beim Speichern von Texten benutzt, wird verwendet. Dadurch lassen sich Texte zwischen beiden Progammen austauschen. Verschiedene Schriftarten werden bereits auf dem Bildschirm angezeigt! Das Handbuch erläutert die Bedienung des Textverarbeitungsprogramms knapp, aber ausreichend. Die Anpassung an verschiedene Druckertypen wird ausführlich beschrieben.

Wenn Sie in Maschinensprache programmieren, leistet Ihnen der leistungsstarke Assembler wertvolle Hilfe. Der Befehlssatz ist zum Assembler »Macro-80« von Microsoft kompatibel, nur Makros verarbeitet er nicht. Das Handbuch geht auf über 50 Seiten intensiv auf die Programmierung des Assemblers ein.

Ein Programm, das sich Z80-Tester nennt, arbeitet wie ein komfortabler Maschinensprache-Monitor. Zusätzliche Befehle erlauben die Kommunikation mit Speichern und Ports (Tore zu Erweiterungen) und bieten softwaremäßig die Grundlagen zur Programmierung von EPROMs. Die Nachrüstung eines EPROM-Programmiergeräts wird damit unterstützt.

Ein kompaktes Terminalprogramm sorgt für die serielle Datenübertragung. Dieses Programm verhält sich kompatibel zu »Modem7« und »XModem« und erlaubt bei Nachrüstung der seriellen Schnittstelle RS232C die Kommunikation mit Mailboxen und anderen Computern.

Der Hersteller betont, daß das gesamte EPROM nur 24 Byte im RAM des CPC belegt, so daß der Arbeitsspeicher nicht wesentlich verkleinert wird. Neue EPROM-Versionen erhält der Anwender gegen eine Aufwandsentschädigung zugeschickt. Erweiterungen mit eingelöteten EPROMs kann man einsenden und neu programmieren lassen.

## Friedliche Aufrüstung

Für jeweils 110 Mark rüstet Philosoft das EPROM-Programmiergerät, beziehungsweise die serielle Schnittstelle, nach. Die Dokumentation zu der Erweiterung beschreibt, wie der Anwender die Schaltung selbst ausbauen kann. Ein Bestückungsplan, Bauteillisten und Angaben zur Einstellung der Jumper sichern dem Bastler den Erfolg beim Aufbau. Die Garantie erlischt allerdings beim Selbstbau, und Kenntnisse im Löten von ICs sind unbedingt erforderlich.

Das EPROM-Programmiergerät erlaubt die Programmierung der EPROM-Typen 2716 bis 27256. Der gewählte EPROM-Typ wird hardwaremäßig über Jumper und auf der Softwareseite über den SELECT-Befehl im Z80-Tester eingestellt. Die Programmierspannung von 12,5 beziehungsweise 21 Volt erzeugt ein Spannungsregler auf der Platine. Dadurch erübrigt sich eine externe Spannungsversorgung.

Einen gewichtigen Nachteil des EPROM-Programmiergeräts macht aus, daß ein Sockel zum Anschluß eines neugebrannten EPROMs an den CPC fehlt. Entweder baut sich der Benutzer eine eigene Schaltung zum Anschluß des EPROMs als Erweiterungs-ROM, oder er kauft eine ROM-Modulbox, in die er den Baustein nur noch einzusetzen braucht.

Die serielle Schnittstelle RS232C ist vollständig kompatibel zur seriellen Schnittstelle von Schneider. Der Anschluß des Kabels zur Datenübertragung erfolgt über eine 25polige SUB-D-Buchsenleiste (Standardanschluß einer RS232C).

Die maximale Baudrate beträgt 31 250 Baud. Der Baudraten-Wert für den Schnittstellenbaustein berechnet sich, indem der Wert 31 250 durch die gewünschte Übertragungsrate geteilt wird. Das Ergebnis dieser Division kann jedoch nur ganzzahlig sein, so daß bei Baudraten über 2400 Baud, die kein ganzzahliges Vielfaches haben, das annähernd 31 250 beträgt, größere Abweichungen entstehen.

Zu der Erweiterung wird kein Gehäuse geliefert. Den Freak wird das nicht stören, doch jeden ordnungsliebenden Anwender befriedigt diese Lösung nicht. Wer will, kann die Schaltung jedoch ohne viel Aufwand in ein preiswertes Kunststoff-Kleingehäuse einbauen. Die Grundausbaustufe in Verbindung mit einem CPC-Modell ohne Speichererweiterung erlaubt sogar den internen Einbau, indem die Schaltung am Erweiterungsanschluß aufgesteckt und nach innen auf die Computerplatine umgeklappt wird. Darauf muß das Gehäuse nur noch vorsichtig zusammengeschraubt werden.

Alles in allem ist die Philosoft-Erweiterung trotz der erwähnten kleinen Schwächen jedem CPC-Besitzer zu empfehlen, der nicht nur mit seinem Gerät spielen und kleinere Aufgaben erledigen will, sondern professionell Texte verarbeiten, Assembler programmieren und austesten, Datenfernübertragung betreiben sowie EPROMs mit eigener Software herstellen möchte. (ma)

Philosoft, Pariser Platz 2, 8000 München 80, Telefon 089/4482601

# Typenrad-Drucker für Schneider

Knapp 700 Mark kostet der SD15 von Schneider Data. Nicht für jeden Zweck ist ein Matrix-Drucker die beste Lösung. Besonders bei »offiziellen« Schriftstücken, den Brief an das Finanzamt oder der Vereinsverwaltung, macht das Punktmuster nicht den besten Eindruck. Ein Typenrad-Drucker hingegen liefert ein Schriftbild, wie Sie es von einer elektrischen Schreibmaschine her gewohnt sind. Den Computer dahinter erkennt niemand mehr.

Der Drucker besitzt eine serielle und eine parallele Schnittstelle. Damit ist er auch für andere Computer als den Schneider CPC problemlos anzusteuern.

Den größten Wert legte der japanische Hersteller, der das von Schneider Data angebotene Gerät produziert, auf Bedienungskomfort. So lassen sich alle DIP-Schalter, mit denen man zwischen den einzelnen Schnittstellen hin- und herschaltet, ohne Demontage des Gehäuses erreichen. Auch der Farbbandwechsel geht ohne schmutzige Finger vonstatten.




**Ausnahmsweise leise: Typenraddrucker SD15**

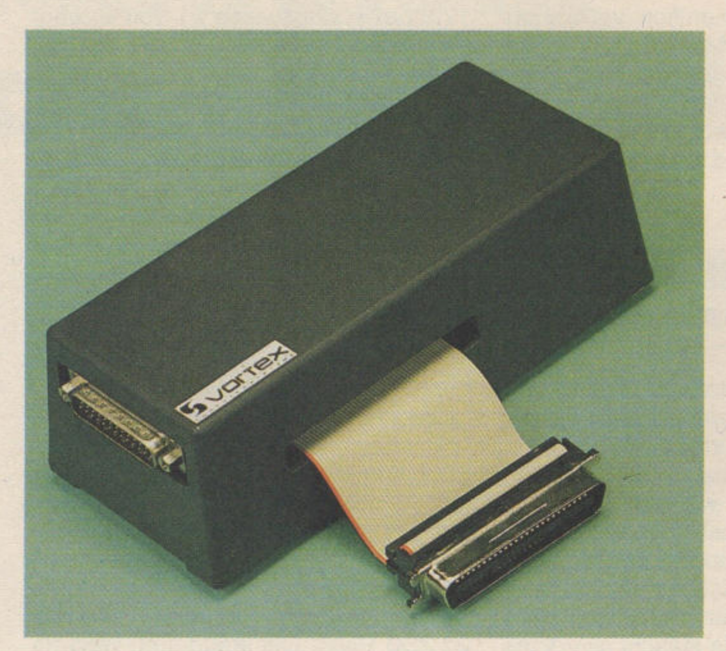
Der SD15 beherrscht alle üblichen Steuercodes, wie Fett- und Schattendruck. Serienmäßig liegt ein deutsches Typenrad bei, andere können aber problemlos eingesetzt werden.

Für ein Typenradgerät ist der SD15 relativ schnell und leise. Allerdings darf man nicht vergessen, daß diese beiden Punkte nicht zu den starken Seiten solcher Geräte zählen. 15 Zeichen pro Sekunde sind hier schon viel. (hg)

Schneider Data, Am Rindermarkt 4, 8050 Freising, Telefon: 08161/2877

# Vortex-Schnittstelle

Mit einem neuen RS232-Interface wartete Vortex vor kurzem auf. Es ist einzeln für 298 Mark zu haben und heißt schlicht »RS-Modul«. In Verbindung mit einem X-Laufwerk des gleichen Herstellers lautet die Bezeichnung »XRS-Modul« und es kostet nur noch 100 Mark Aufpreis gegenüber dem einfachen X-Controller. Genau diese 100 Mark zuzüglich 10 Mark Bearbeitungsgebühr zahlen Besitzer eines X-Laufwerks für den Austausch ihres Controllers gegen die RS232-Version. Die Schnittstelle ist mit einem reichen Vorrat von RSX-Befehlen in das Locomotive-Basic eingebunden. Darüber besteht volle Kompatibilität zur seriellen Schnittstelle von Amstrad (nicht jedoch zum Schneider-Interface, das über keine Basic-Einbindung verfügt), denn deren Befehlswortschatz ist eine Teilmenge der Kommandovielfalt des RS-Moduls. Auf der Maschinensprach-Ebene endet allerdings jegliche Kompatibilität. Auch unter CP/M ist das Interface anzusprechen. Zum Lieferumfang gehört ein Terminalprogramm für dieses Betriebssystem. Eine Übersicht der Befehle finden Sie in der untenstehenden Tabelle. Positiv fiel uns auch der Umfang und die Qualität des deutschsprachigen Handbuchs auf. Es liefert eine so umfassende Information zu diesem Produkt, wie wir sie selten in den Händen hatten. (ja)




**Im Gehäuse des XRS-Moduls findet auch der Controller der Schneider-3-Zoll-Laufwerke Platz**

Vortex Computersysteme, Falterstraße 1 – 5, 7101 Flein, Telefon 07131/52061

| Basic-Befehle zur Schnittstellen-Steuerung | |
|---|---|
| BLOW | ASCII-Datei intelligent senden (siehe Befehl SUCK) |
| BREAKSEND | Break senden, wenn der Sendepuffer leer ist |
| CHANNEL | Kanalwahl (arbeitet nur in Verbindung mit neuer SP plus-Erweiterung oder neuem FDC plus-Controller) |
| CLOSESIO | warten, bis der Sendepuffer leer ist |
| COUNTER | Zähler des Timers direkt einstellen (für exotische Baudraten) |
| CTRLACTION | Steuerzeichen interpretieren (ausführen) |
| CTRLDISPLAY | Steuerzeichen darstellen (als ASCII-Zeichen) |
| FULLDUPLEX | stellt nur die empfangenen Zeichen auf dem Bildschirm dar |
| HALFDUPLEX | stellt auch über die Tastatur eingegebene Zeichen dar |
| INBLOCK | Zeichenkette lesen |
| INCHAR | einzelnes Zeichen lesen |
| INFILE | gelesene Daten als Datei speichern |
| INITSIO | Schnittstelle initialisieren |
| NOXON | XON/XOFF-Protokoll unterdrücken |
| OUTBLOCK | Zeichenkette senden |
| OUTCHAR | einzelnes Zeichen senden |
| OUTFILE | Inhalt einer ASCII-Datei senden |
| PARALLEL | Druckausgabe auf Centronics-Port |
| RINGWAIT | auf »Ring detect«-Meldung warten |

| Basic-Befehle zur Schnittstellen-Steuerung | |
|---|---|
| SETBLOCKEND | Blockende-Kennung definieren |
| SETENDFILE | Dateiende-Kennung definieren |
| SETSIO | RS232-Parameter einstellen |
| SETTIMEOUT | definieren der Zeit für Sendeversuche |
| SERIAL | Druckausgabe über RS232 leiten (für PRINT #8 und LIST #8) |
| SIO | Schnittstellen-Status ermitteln |
| SUCK | Empfang einer ASCII-Datei, die ein anderer CPC mit dem Befehl BLOW sendet |
| TERMINAL | leitet Tastatureingaben zur Sendung auf die Schnittstelle und stellt empfangene Zeichen auf dem Bildschirm dar (zur Datenübertragung) |
| XON | XON/XOFF-Protokoll aktivieren |
| TRANSFER.COM | CP/M-Transferprogramm generieren (arbeitet wie BLOW und SUCK) |
| | **Zusätzliche Befehle** |
| BASE | RAM-Basisadresse eines Erweiterungs-ROMs ermitteln |
| ROMCAT | Katalog aller angeschlossenen ROMS |
| ROMOFF | Alle oder einzelne Hintergrund-ROMs ausschalten |
| STATUS | Basic-Programmstatus ausgeben (HIMEM, Programmstart, -ende, -länge, verfügbarer Speicherplatz und Zeichendefinition) |
| VERSION | Versions-Nummer und Updating-Datum des Interface-ROMs ausgeben |

**Tabelle. Das Vortex-RS-Modul bietet eine Vielzahl Befehle von der Parametereinstellung bis hin zur Übertragung kompletter Dateien**









## Top-Listings dieser Ausgabe:

**Anwendungen:**
**Datafine:** Eine relative Dateiverwaltung mit tollen Leistungsmerkmalen.
**Termin:** Verwalten Sie Ihre täglichen Termine einfach mit dem Computer.
**Giro:** Immer im Bilde über den Stand des Girokontos. Um regelmäßige Buchungen brauchen Sie sich dabei nicht zu kümmern.
**Backup:** Das einzige wirklich universelle Kopierprogramm für alle Speichermedien. Auch der Transfer vom einen zum anderen ist kein Problem.

**Spiele:**
**Softchef:** Führen Sie Ihre eigene Softwarefirma in die Gewinnzone, indem Sie die richtigen Programme einkaufen und geschickt vermarkten.

**Utilities:**
**Scale:** Bildschirm-Textausgaben in jeder beliebigen Größe.
**Kompex:** Der Kompressor für alle Speicherinhalte hilft Ihnen, kostbaren Speicherraum einzusparen.
**Disk Plus:** Neue Befehle erlauben vielfältige Manipulationen der Disketten.

**1 Diskette für Schneider-Computer**
**Bestell-Nr. 25713** (sFr 29,50/öS 349,-*) **DM 34,90***

## Weitere Stammhefte zum Thema Schneider-Computer

### Happy-Computer, Ausgabe 12/86

**Goldrain.** Wertet Ihre Spielkarten des Bild-Goldregen-Spiels aus. **Screen-Compressor.** Speichert Bildschirminhalte platzsparend und mit erheblichem Geschwindigkeitsgewinn. Sie haben dabei die Wahl zwischen ganzen Bildschirmen, Ausschnitten und Windows. **Kursiv.** Ideal für Textverarbeitung: Verwenden Sie auf dem Bildschirm denselben kursiven Zeichensatz wie auf dem Drucker. **Super-CLS.** Neuer RSX-Befehl zur effektvollen Bildschirmlöschung. **Newgosub.** Ein Patch des GOSUB-Befehls erlaubt strukturierte Basic-Programmierung mit Unterprogrammnamen (nur CPC 464). **DEC$-Patch.** Endlich die perfekte Abhilfe für einen Fehler im Basic-Interpreter des CPC 464: Die Syntax des Befehls DEC$ ist nun korrigiert und somit kompatibel zu den beiden anderen CPC-Modellen (nur CPC 464). **Public-Domain.** Als besonderen Leckerbissen bieten wir Ihnen verschiedene Public-Domain-Programme. Darunter finden Sie je einen Interpreter der KI-Sprachen Lisp und Prolog mit Dokumentation und Beispielen sowie einen Forth-Compiler und einen Makroassembler.

**1 Diskette für Schneider-Computer**
**Bestell-Nr. LH 8612 SD**
**DM 34,90*/sFr 29,50/öS 349,-***

**2 Kassetten für Schneider-Computer**
**Bestell-Nr. LH 8612 SK**
**DM 34,90*/sFr 29,50/öS 349,-***

### Happy-Computer, Ausgabe 11/86

**Dash.** Listing des Monats in Ausgabe 11. Mit fantastischen Spielmöglichkeiten und eigenem Spielfeldeditor. **Quadrat-Killer.** Taktisches Spiel mit völlig neuer Grundidee für zwei Personen und mit drei verschiedenen Varianten (10/86). **Giro.** Behalten Sie mit Ihrem CPC den Stand Ihres Girokontos im Auge (11/86). **Modem.** Terminalprogramm zur Datenfernübertragung mit Akustikkoppler (10/86). **Tapemonitor.** Dieses Monitorprogramm verhilft Ihnen zum Einblick in Daten, die auf Kassettenband gespeichert sind. Teilweise zerstörte Dateien lassen sich damit wieder restaurieren (10/86). **Disc-RSX.** Neue RSX-Befehle erweitern Ihren Zugriff auf Disketteninhalte. Das Demonstrationsprogramm erweitern Sie leicht beispielsweise zum leistungsstarken Kopierprogramm für geschützte Diskettenprogramme (10/86). **Unerase.** Macht versehentlich mit ERA gelöschte Diskettendateien menügesteuert per Tastendruck wieder zugänglich (10/86).

**1 Diskette für Schneider-Computer**
**Bestell-Nr. LH 8611 SD**
**DM 34,90*/sFr 29,50/öS 349,-***

**1 Kassette für Schneider-Computer**
**Bestell-Nr. LH 8611 SK**
**DM 34,90*/sFr 29,50/öS 349,-***

* inkl. MwSt. Unverbindliche Preisempfehlung.

## Programme aus früheren Happy-Ausgaben

| Ausgabe | Thema | Bestell-Nr. | | DM | sFr | öS |
|---|---|---|---|---|---|---|
| 12/86 | Schneider | LH 8612 SD | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 8612 SK | 2 Kassetten | 34,90* | 29,50 | 349,-* |
| 11/86 | Schneider | LH 8611 SD | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 8611 SK | Kassette | 34,90* | 29,50 | 349,-* |
| 9/86 | Schneider | LH 8609 SD | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 8609 SK | Kassette | 34,90* | 29,50 | 349,-* |
| 7/86 | Schneider | LH 8607 SD | Diskette | 34,90* | 29,50 | 349,-* |
| 4/86 | Schneider | LH 8604 SD | Diskette | 29,90* | 24,90 | 299,-* |
| | | LH 8604 SK | Kassette | 29,90* | 24,90 | 299,-* |
| 12/85 | Schneider | LH 8512 D | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 8512 G | Kassette | 29,90* | 24,90 | 299,-* |

## Programme aus früheren Happy-Sonderheften

| Ausgabe | Thema | Bestell-Nr. | | DM | sFr | öS |
|---|---|---|---|---|---|---|
| 10/86 | Schneider | LH 86S10 D | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 86S10 K | 2 Kassetten | 34,90* | 29,50 | 349,-* |
| 7/86 | Schneider | LH 86S7 SD | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 86S7 SK | Kassette | 34,90* | 29,50 | 349,-* |
| 4/86 | Schneider | LH 86S4 D | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 86S4 K | Kassette | 29,90* | 24,90 | 299,-* |
| 1/86 | Schneider | LH 86S1 D | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 86S1 K | Kassette | 29,90* | 24,90 | 299,-* |
| 2/85 | Schneider | LH 85S2 D | Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 85S2 V | 5¼"-Diskette | 34,90* | 29,50 | 349,-* |
| | | LH 85S2 K | Kassette | 29,90* | 24,90 | 299,-* |

**Einige Tips zum Umgang mit den Leserservice-Disketten:**

Auf der Kassette und Diskette zu dieser Ausgabe finden Sie ein Basic-Programm namens »README.BAS«. Da es am Anfang der Kassette 1 gespeichert ist, starten Sie es bitte zuerst. Sie erhalten dadurch Informationen über die enthaltenen Programme. Dort erfahren Sie zu jeder Datei, was sie bewirkt und wo der gedruckte Beitrag dazu in der Ausgabe zu finden ist.
Bei früheren Ausgaben hieß dieses Inhaltsverzeichnis ebenso beziehungsweise »LISTME.BAS«. Dort besteht es aus einer ASCII-Datei, die Sie mit »LOAD "README"« im normalen Locomotive-Basic laden und durch »LIST« auf den Bildschirm beziehungsweise mit »LIST #8« auf dem Drucker ausgeben.

**Bestellungen bitte an:** Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. **Schweiz:** Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656. **Österreich:** Ueberreuter Media Handels- und Verlagsgesellschaft mbH, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0, Microcomput-ique E. Schiller, Fasangasse 21, A-1030 Wien, Telefon (0222) 785661, Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. **Bestellungen aus anderen Ländern bitte nur schriftlich an:** Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die eingeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

# Original-dokumentation für Schneider

Digital Research hat sein Archiv geplündert. In der Reihe Originaldokumentationen von Markt&Technik gibt es jetzt die Programmier- und Benutzerhandbücher für CP/M 2.2 und CP/M Plus. In diesen Büchern findet man alle Informationen über Diskettenzugriffe, Speicherverteilung, BDOS-Funktionen und so weiter, die Digital Research den Software-Entwicklern zur Verfügung stellt. Damit sind die Betriebssystem-Handbücher sicher nicht für den Normalbürger gedacht, insbesondere, da sie überwiegend als Nachdruck der englischen Originalausgaben vorliegen. Allein unter CP/M Plus ist das Benutzerhandbuch auf Deutsch erschienen.

Utilities und Informationen über den SID finden Sie in dem »Programmer's Utilities und SID für CP/M 2.2 und CP/M 3.0 (Plus) in englischer Sprache« (so der etwas umständliche Titel). Wenn Sie diese 300 Seiten auch noch gelesen haben, macht Ihnen unter CP/M niemand mehr etwas vor.

Auch für Grafik-Fans mit einem CPC 6128 oder Joyce und für alle Logo-Fans ist in dieser Reihe etwas dabei. Das GSX-Handbuch stellt in gleicher Weise sämtliche Grafik-Routinen vor. Im »Dr. Logo«- Benutzerhandbuch finden Sie eine Einführung in Logo und ein Lexikon mit allen Logo-Anweisungen. Es gibt ja schon eine Menge Bücher über das Logo auf den Schneider-Computern. Mit dem Buch von Digital Research bekommen Sie aber erstmals alles auf einen Blick über Turtle-Grafik, Prozeduren, Rekursionen und so weiter. (hg)

Reihe: Originaldokumentation, Verlag Markt&Technik, Haar bei München
»Das Handbuch des CP/M 2.2-Betriebssystems«, 38 Mark, ISBN 3-89090-369-X
»CP/M Plus Betriebssystem«, 38 Mark, ISBN 3-89090-371-1
»Programmer's Utilities und SID für CP/M 2.2 und CP/M 3.0 (Plus)«, 49 Mark, ISBN 3-89090-372-X
»GSX-Handbuch«, 39 Mark, ISBN 3-89090-373-8
»Dr. Logo Benutzerhandbuch«, 42 Mark, ISBN 3-89090-116-6



**Informationen vom Entwickler: Originaldokumentationen zu CP/M**

# dBase II voll ausgenutzt

Fakturierung heißt ein Programm, das die tägliche Büroarbeit auch mit einem CPC 6128 oder einem Joyce minimiert. Speziell Angebot- und Rechnungschreiben wird mit der 94 Mark teuren Software ein »Klacks«. Das Hauptaugenmerk wurde auf EDV-Laien gelegt. Kundenanschreiben und Stammdatenergänzung sind ein Arbeitsgang. Daher ist eine Extra-Erfassung neuer Kunden unnötig.

Da das Programm auf der Diskette im Quellcode vorliegt, ist es sehr einfach, die Arbeitsweise an individuelle Bedürfnisse anzupassen. Aus dem Angebot wird so schnell eine Rechnung und aus dem Gesamtprogramm eine Schallplattenverwaltung.

Das deutsche Handbuch umfaßt zirka 200 Seiten. Damit ist gewährleistet, daß Sie auch wirklich mit der Fakturierung klarkommen. Des weiteren lernen Sie dBase II sehr genau kennen – beinahe so gut wie mit einer (bis heute noch nicht preiswert erhältlichen) Lerndiskette. (hg)

Markt&Technik AG, Hans-Pinsel-Str 2, 8013 Haar bei München, Tel: 089/4613-0

# Das große CPC-Arbeitsbuch

Jeder, der mehr über seinen Schneider CPC – egal ob 464, 664 oder 6128 – wissen will, der kommt um das neue Buch aus dem Franzis-Verlag nicht herum. 68 Mark kosten die zirka 450 Seiten voll mit Informationen. Zu Anfang kommt in wenigen Worten die Hardware des Computers zur Sprache. Dieser Teil unterscheidet sich allerdings in nichts von den vielen anderen Büchern über die CPC-Computer.

Interessanter wird es bei den Informationen über den Speicheraufbau, die Variablenablage und die Struktur der Basic-Listings. Bisher war dieses Wissen nur hübsch verteilt in verschiedenen Zeitschriften und Büchern zu lesen. Eine Zusammenfassung fehlte bislang. Gleiches gilt für den Bildschirmaufbau, der ja beim Schneider nicht ganz einfach zu verstehen ist. Das CPC-Arbeitsbuch deckt mit seiner Zusammenfassung jetzt diese Lücke ab.

Alle Kapitel enthalten Programme, mit denen man das gerade Erlernte umsetzen kann. Viele Listings eignen sich dabei auch für die tägliche Arbeit, so zum Beispiel die Sprite-Routine und das Programm REM-Kill.

»Das große CPC-Arbeitsbuch« ist für jeden, der mehr als nur fertige Software laufen lassen will, ein absolutes Muß. Leider wird der sprachliche Stil und auch die äußere Aufmachung als Weichcover-Buch dem Inhalt nicht gerecht. Aber über diese »Hardware-Schwäche« kann man hinwegsehen. (hg)

»Das große CPC-Arbeitsbuch«, Miedel/Kotulla, Franzis-Verlag München, zirka 450 Seiten, 68 Mark, ISBN 3-7723-8421-8

# Eisenbahnspielen mit dem Schneider

Für 10 Mark gibt es bei Märklin eine Diskette für den Schneider CPC zur Demonstration der digital gesteuerten Eisenbahn. Sinn und Zweck der Programme ist es, die Grundlagen einer computergesteuerten Modellbahnanlage zu zeigen. Wußten Sie, daß mit

```
1000 INPUT "Loknummer (1-80):",LN
1010 INPUT "Geschw.   (0-15):",GE
1020 WAIT &F8EE,4
1030 OUT &F8EF,GE
1040 OUT &F8EF,LN
1050 GOTO 1000
```

bis zu 80 Lokomotiven mit unterschiedlichen Geschwindigkeiten über Ihre Anlage gesteuert werden können? Voraussetzung ist neben dem Computer eine serielle Schnittstelle und das Interface zu der Digital-HO-Mehrzugsteuerung. Die Adresse im vorhergehenden Beispiel ist von der Schnittstelle abhängig – hier wird die RS232 von Schneider angesprochen.

Die 21 Programme auf der Diskette geben einen interessanten Einblick, was mit einer Modellbahnanlage und einem Computer alles machbar ist. Von einer Uhr für Modellbahnzeit bis hin zu einem Gleisplan mit Besetztanzeige ist alles vorhanden. (hg)

Geb. Märklin & Cie. GmbH, Stuttgarter Str. 55-57, 7320 Göppingen, Tel.: 07161/608-0

## Hardcopies par excellence

»Copyshop« von DMV ist ein komfortables Hardcopy-Programm, das mit allen gängigen Druckern zusammenarbeitet. Gleich zu Anfang fällt positiv auf, daß Copyshop mit keinerlei Kopierschutz versehen ist. Im Gegenteil, man ermuntert den Käufer sogar dazu, Copyshop seinen Bedürfnissen anzupassen und einzelne Routinen in seinen eigenen Programmen zu verwenden.

Daß Copyshop mehr als eine Hardcopy-Routine ist, sieht man spätestens nach dem Programmstart am Menü. Acht Funktionen, wie Bilder laden und speichern, Modus-Einstellung und Wahl des Druckformats stehen dem Anwender zur Auswahl.

In der Funktion »Editor« läßt sich ein Bild bildpunktweise manipulieren und mit Text versehen. Eine Fill-Funktion und eine Bildverschiebe-Routine erlauben umfangreiche Änderungen. Auf Knopfdruck wird der gesamte Bildschirminhalt invers dargestellt.

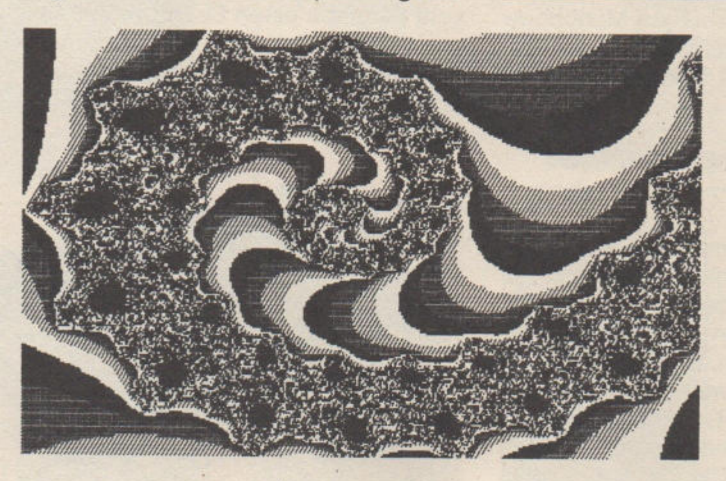
Der Clou ist die Funktion »Farben/Muster«, die dem Anwender eine Auswahl an Mustern anbietet. Jeder Bildschirmfarbe kann er ein Muster zuordnen. Auf diese Weise entstehen auch von mehrfarbigen Bildern eindrucksvolle Hardcopies, indem jeweils ein bestimmtes Muster eine einzelne Farbe repräsentiert.

Die im Programm voreingestellte Zuordnung von Farben und Mustern ist recht sinnvoll, und der Ausdruck des mitgelieferten Demobildes liefert ein gutes Beispiel für die Fähigkeiten des Programms. Vor dem Ausdruck kann man zusätzlich entscheiden, ob das gesamte Bild oder eine Ausschnittsvergrößerung gewünscht ist. Die beigefügte Dokumentation reicht mit 20 Seiten aus, um alle Funktionen von Copyshop zu erklären.

Nach so viel Lob auch etwas Tadel: Zum einen funktionieren auf den meisten Druckern zwei der vier Druckformate nicht richtig, da Copyshop sinnlose Zeichen am Bildrand ausgibt und zum Teil auch ins Bild einstreut.

Der zweite Nachteil von Copyshop liegt in den fehlenden Anpassungshinweisen für gebräuchliche Druckertypen. Es existiert zwar ein spezieller Modus zum Einstellen der Escape-Sequenzen, aber die angezeigten Werte gelten nur für den Schneider NLQ 401. Die Daten für den eigenen Drucker muß sich der Anwender selbst zusammensuchen. In der Redaktion haben wir beim Testen folgende Anpassungen herausgefunden: Besitzer von Epson-Druckern müssen als erste Steuersequenz die Zahlenfolge »27,64,27,49,27,108,06« eingeben. Für die zweite Steuersequenz sind die Werte »13,10,27,76,32, 03« erforderlich. Bei Benutzung eines Star-Druckers tauschen Sie lediglich den Wert 108 gegen 77 aus. (ma)

DMV Daten & Medien Verlagsgesellschaft, Postfach 250, Fuldaer Str. 6, 3440 Eschwege



Eine eindrucksvolle Demonstration von Copyshop

## Lesefutter zum Schneider PC

Das Buch »Der Schneider PC« ist eines der ersten Bücher zu diesem Thema. Schon nach den ersten Seiten stellt der Leser fest, daß das Buch in Windeseile produziert wurde, denn anscheinend verzichtete man aus Zeitgründen auf jede Art von sprachlicher Überarbeitung. Eine Schlußredaktion hätte dem Buch sicherlich gutgetan. So aber hat der Leser fast 360 Seiten lang holpriges Deutsch und flapsige Bemerkungen zu ertragen.

Dieses Manko sollte aber nicht vergessen lassen, daß das Buch inhaltlich und fachlich einwandfrei gelungen ist. Auch die intensive Suche nach kleinen oder größeren Fehlinformationen und Druckfehlern blieb ohne Erfolg.

Zu Beginn erhält der Leser einen Einblick in die Hardware des Schneider PC. Der Autor geht allerdings nicht ins Detail, da auf diesem Gebiet bei den meisten Lesern nur begrenztes Interesse besteht. Wer eine Aufschlüsselung der Hardware-Struktur des PC erwartet, wird von der oberflächlichen Exkursion enttäuscht sein.

Darauf widmet sich das Buch ausführlich den zum Schneider PC mitgelieferten Betriebssystemen MS-DOS und DOS Plus. Parallelen wie Unterschiede arbeitet der Autor sauber heraus und stellt sie einander gegenüber. Wenn auch der MS-DOS-Teil nur für Neulinge interessant ist, so bietet die Gegenüberstellung zu DOS Plus auch dem erfahreneren MS-DOS-Anwender neue Informationen und zeigt Stärken wie Schwächen auf.

Einige Kritikpunkte hätte man sich jedoch etwas deutlicher herausgestellt gewünscht. Bloße Andeutungen sind manchmal nicht ausreichend.

Anschließend folgt die Behandlung der mausgeführten grafischen Benutzeroberfläche GEM, die ebenfalls im Lieferumfang des Schneider PC enthalten ist. Auch wenn GEM von allen Seiten als leicht erlernbar gepriesen wird, zeigen doch die umfangreichen Erklärungen, daß etwas Aufwand nötig ist, um es endgültig zu beherrschen. Eine Vielzahl von Hardcopies erleichtert dem Leser die Orientierung und das Verständnis. Alle Beispiele und Übungen sind so angelegt, daß sie der Leser unmittelbar nachvollziehen kann.

Die Fähigkeiten von GEM Paint, das unter GEM läuft, führt der Autor kurz als Demonstration vor. Das Basic 2 des Schneider PC wird nur gestreift. Hier verweist der Autor auf die nicht vorhandenen ausführlichen Informationen des Handbuchs.

Lobenswert ist der Anhang, der alle Schlüsselwörter und Befehle mit Kurzdefinitionen übersichtlich aufführt. Dieser Teil ist zum Auffrischen der Kenntnisse und zum Nachschlagen unentbehrlich.

Zusammenfassend kann das Buch als ausgezeichnetes Lehrbuch für den Anfänger gelten. Der übersichtliche Anhang bringt auch dem fortgeschrittenen Computerbenutzer wertvolle Informationen, dem Profi allerdings nichts Neues. (ma)

Rudi Kost, »Der Schneider PC«, Markt & Technik Verlag, ISBN 3-89090-415-7, Preis: 49 Mark



# Nur halb geknackt

**Die Sensation ist perfekt: Zwei unscheinbare Kästchen machen das Heer von Knackern arbeitslos, die Software bislang per mühevoller Handarbeit »knackten«.**

Was zunächst wie eine Hiobsbotschaft für die Software-Industrie klingt, trifft bei genauerem Hinsehen doch nicht zu. Denn die beiden Kopiermodule, die hier zum Zweikampf antreten, vermögen zwar fast beliebig Programme zu vervielfältigen, die Kopien sind aber zur Freude der Programm-Entwickler wiederum nur mit dem jeweiligen Modul lauffähig. Einer der Kontrahenten war bereits in der Happy-Computer, Ausgabe 11/86, Gegenstand einer näheren Betrachtung. Der Mirage »Imager« hatte sich damals den Ruf eines Kopierschutz fressenden »Byte-Munchers« erworben und die Versprechungen seines Herstellers mit Bravour gehalten.

## Ungleiche Brüder

Inzwischen steht mit dem »Multiface Two« ein starker Konkurrent Gewehr bei Fuß. So ähnlich die Zielsetzung, so unterschiedlich ist die praktische Umsetzung geraten. Der Imager steckt in einem nur etwas mehr als Zigarettenschachtel großen Gehäuse, das sich optisch und mechanisch nahtlos an den Computer anfügt. Die Behausung des Multiface fiel noch etwas kleiner aus, beansprucht aber trotzdem auf dem Computertisch mehr Platz, weil es dort an einem kurzen Kabelstück lose herumliegt. Beide arbeiten am Erweiterungsport und verfügen zum Betrieb weiterer Peripherie über einen durchgeführten Bus. Ebenfalls gemeinsam ist ihnen die rote Taste zum Erwecken ihrer schlummernden Talente. Ohne diesen Knopfdruck ahnt der Computer nämlich nichts von ihrer Anwesenheit. Auf dem Multiface fällt eine zweite, grüne Taste ins Auge. Ihre Funktion vermißte so mancher CPC-Besitzer sicher schon des öfteren: Sie löst einen Reset aus (beim Imager gibt es eine ähnliche Funktion im Menü). Um nun Programme zu kopieren, verfährt man zunächst wie gehabt. Erst wenn sie ganz normal geladen und eventuelle Schutzabfragen (Farbcodes, Lenslock oder andere) passiert sind, löst ein Druck des roten Knopfs den »verhängnisvollen« Vorgang aus. Beim Imager erscheint nun in den oberen beiden Bildschirmzeilen ein Menü. Das Multiface nutzt dafür die untersten zwei Zeilen. Der wichtigste Menüpunkt ist sicher <S> für »Save«. Einigkeit, auch was die Speichermedien betrifft: In beiden Fällen besteht die Wahl zwischen dem Recorder (mit zwei verschiedenen Geschwindigkeiten) und dem Diskettenlaufwerk. Wichtig ist hier, daß sowohl der Imager als auch das Multiface auf dem originalen Amstrad/Schneider-Controller bestehen und sich keinesfalls mit Fremdlaufwerken zufriedengeben. Das Multiface geht gar noch einen Schritt weiter und verschmähte beim Test die Zusammenarbeit mit einem CPC mit eingebauter Speichererweiterung. Schade, daß die ten. Nun wäre eigentlich der Test bereits beendet, gäbe es da nicht noch ein paar Aspekte zu bedenken – positive und negative.

## Spielkameraden

Die Module speichern nicht im Sinne des Wortes Programme, sondern Speicherinhalte. Sie untersuchen also den gesamten Arbeitsspeicher des Computers auf Veränderungen gegenüber dem Einschaltzustand. All diese sammeln sie und legen sie auf Kassette oder Diskette ab. Natürlich gehören dazu auch Farbinformationen, Registerinhalte des Prozessors und derlei mehr.




**Links: Der Mirage Imager war das erste Gerät seiner Gattung für die CPC-Serie. Rechts: Gnade der späten Geburt: Neuere Geräte vermeiden oft die Fehler ihrer Vorbilder. So auch das Multiface Two. Es bietet mehr Fähigkeiten und Komfort.**

Testkandidaten so wählerisch sind. Ansonsten kann man mit ihnen richtig glücklich werden, denn ihre Daseinsberechtigung stellen beide wirkungsvoll unter Beweis. Keines der Spiele, die wir von Kassetten auf Disketten transferierten, stellte einen der beiden Probanden vor Schwierigkeiten. Lediglich Programme, die Programmteile, Daten oder Bilder nachladen, sind auf diese Weise nicht zu verarbeiten. Was die Wege zum

## Sieger nach Punkten

Erreichen des Ziels betrifft, herrscht indes Uneinigkeit. Während die Speicher- und Ladevorgänge mit dem Imager jedesmal zu einer unfreiwilligen Kaffeepause zwingen, gibt sich das Multiface in dieser Disziplin mehr Mühe, denn es erledigt derlei Datentransporte in weniger als einem Viertel der Zeit. Damit ist es Rundensieger nach Punk-

Daraus ergeben sich nicht zu unterschätzende Vorteile: Spiele lassen sich an jeder beliebigen Stelle unterbrechen. Der gespeicherte Spielstand dient dann dazu, später von dieser Stelle an weiterzuspielen, ohne wieder die ganze Vorarbeit zu leisten. Aber auch andere Perspektiven eröffnen sich. Entsprechende Menüpunkte erlauben einfache Eingriffe und Veränderungen in Programmen. So sind mit beiden Modulen die Farben der Bildschirmdarstellung frei veränderbar, so daß Besitzer eines grünfarbenen Monitors die farbige Grafikdarstellung einiger Programme an ihr Sichtgerät anpassen können. An Manipulationen kopierter Programme hat das Multiface jedoch noch erheblich mehr zu bieten. Es beherbergt einen Maschinencode-Monitor, mit dessen Hilfe sich beliebige Speicher- und Registerinhalte und diverse andere Statusinformationen anzeigen und verändern lassen. Dazu

# Bücher zu Schneider-Computern

J. Höckstädt
**Textverarbeitung mit LocoScript**
**1986, 246 Seiten**
Dieses Buch führt Schritt für Schritt in die Arbeit mit diesem Textsystem ein.
Beginnend mit dem Erfassen, Korrigieren und Speichern von Texten, über das Verschönern des Schriftbilds bis hin zum Serienbrief und dem Einsatz von Fremddruckern erfährt der Anwender noch vieles mehr.
**Best.-Nr. MT 90198**
**ISBN 3-89090-198-0**
**DM 39,–/sFr. 35,90/öS 304,20**

Digital Research
**Das Handbuch des CP/M-2.2-Betriebssystems**
**1986, 342 Seiten**
Alles über CP/M 2.2 von seinen Entwicklern: Die besten Informationen über dieses 8-Bit-Standard-Betriebssystem.
**Best.-Nr. MT 90369**
**ISBN 3-89090-369-X**
**DM 38,–/sFr. 35,–/öS 296,40**







R. Kost
**Der Schneider PC**
**1986, 354 Seiten**
Der Schneider PC ist eine Herausforderung an die Welt der Mikrocomputer. Mit seinen zwei Betriebssystemen, dem bewährten MS-DOS, mit dem die Tür zu den Kompatiblen offensteht, und dem neuentwickelten DOS Plus, das auch den Zugriff auf die CP/M-86-Software ermöglicht, ist der Schneider PC universell einsetzbar. Beide Betriebssysteme werden anschaulich beschrieben und dokumentiert.
Schneiders besonderer Service als Dreingabe ist die bedienerfreundliche Benutzeroberfläche mit den speziell angepaßten Desktop-Utensilien und dem Paint-Programm aus der GEM-Palette. Wie sie funktionieren, wie man mit ihnen arbeitet und ihre Möglichkeiten ausschöpfen kann, bildet den Hauptteil dieses Handbuchs. Der Anwender wird schrittweise in GEM eingeführt und auch mit den anderen GEM-Produkten bekannt gemacht.
**Best.-Nr. MT 90415**
**ISBN 3-89090-415-7**
**DM 49,–/sFr. 45,10/öS 382,20**
















C. Straush
**Schneider CPC Grafik-Programmierung**
**1986, 231 Seiten**
Dieses Buch wendet sich an die Schneider-CPC-Besitzer, die alles über die Grafikfähigkeiten ihres Computers wissen wollen. Es bietet einen umfassenden Überblick über die verschiedenen Anwendungsbereiche der Grafikprogrammierung: zwei- und dreidimensionale Diagrammdarstellungen, Definition und Bewegung von Sprites, Entwurf von Titelgrafiken, Einsatz der Grafik bei der Unterstützung anderer Programme. Alle Beispiele auf Diskette (Best.-Nr. MT 898, DM 34,90*) und Kassette (Best.-Nr. MT 873, DM 29,90*).
**Best.-Nr. MT 90182**
**ISBN 3-89090-182-4**
**DM 46,–/sFr. 42,30/öS 358,80**
* inkl. MwSt.
Unverbindliche Preisempfehlung

J. Hückstädt
**Der Schneider CPC 6128**
**1985, 273 Seiten**
Dieses Buch ist für jeden CPC 6128-Besitzer eine wertvolle Hilfe, die vielfachen Möglichkeiten dieses bisher einmaligen Computers kennenzulernen und anzuwenden. Der Computerneuling wird Schritt für Schritt in den Umgang mit dem Computer und in die BASIC-Programmierung eingeführt, bis er alle notwendigen Kenntnisse besitzt, die mancher Profi bereits mitbringt. Aber an dieser Stelle wird das Programmieren mit dem CPC 6128 erst interessant, nämlich dann, wenn es darum geht, eine eigene Dateiverwaltung aufzubauen oder Grafik und Sound zu programmieren. Weiterhin erfahren Sie alles über CP/M Plus auf dem CPC 6128.
**Best.-Nr. MT 90192**
**ISBN 3-89090-192-1**
**DM 46,–/sFr. 42,30/öS 358,80**

J. Hückstädt
**CP/M-Plus-Anwender-Handbuch CPC 6128/Joyce**
**1986, 256 Seiten**
Mit der Verfügbarkeit von CP/M-Plus steht den Besitzern von Schneider-Heimcomputern der Zugang zu einer der vielleicht größten Software-Bibliotheken der Welt offen. Mit Hilfe dieser Programme kann die Grenze vom bloßen Heim- zum Personalcomputer überschritten werden.
Sie erfahren alles über die Organisation der Dateien, die Grundlagen der Assemblerprogrammierung sowie über den Aufbau von CP/M-Plus.
● Dieses Buch gehört zu den ersten in deutscher Sprache, die die weiterentwickelte Version 3.0 (CP/M-Plus) ausführlich behandeln.
**Best.-Nr. MT 90197**
**ISBN 3-89090-197-2**
**DM 46,–/sFr. 42,30/öS 358,80**

H. Tischer
**Programmentwicklung unter CP/M 2.2 auf dem CPC 464/664**
**1986, 340 Seiten**
Dieses Buch vermittelt alle Informationen, die zum selbständigen Entwickeln von CP/M 2.2-Programmen nötig sind. Besprochen wird sowohl die grundlegende Funktionsweise des CP/M Betriebssystems als auch alle dem Anwender schon zur Verfügung stehenden Systemroutinen, die diesem viel Arbeit ersparen. Zwei Kapitel beschäftigen sich dabei ausschließlich mit den zusätzlichen Möglichkeiten, die nur die Computer CPC 464/664 bieten.
Kenntnisse der 8080- oder Z80-Assemblersprache sind erforderlich.
**Best.-Nr. MT 90209**
**ISBN 3-89090-209-X**
**DM 52,–/sFr. 47,80/öS 405,60**

C. Straush
**CPC 464 – Programmieren in Maschinensprache**
**1985, 276 Seiten**
Dieses Buch weiht in die Arbeitsweise des BASIC-Interpreters ein und erklärt die Funktionsweise der Bauteile des Geräts und deren Zusammenwirken.
**Best.-Nr. MT 829**
**ISBN 3-89090-166-2**
**DM 46,–/sFr. 42,30/öS 358,80**

Dr. P. Albrecht
**MULTIPLAN für den Schneider CPC**
**1985, 226 Seiten**
**Best.-Nr. MT 835**
**ISBN 3-89090-186-7**
**DM 49,–/sFr. 45,10/öS 382,20**

G. Jürgensmeier
**WordStar 3.0 mit MailMerge für den Schneider CPC**
**1985, 435 Seiten**
Das unentbehrliche Zusatz-Handbuch für die Arbeit mit dem Schneider CPC.
**Best.-Nr. MT 779**
**ISBN 3-89090-180-8**
**DM 49,–/sFr. 45,10/öS 382,20**

Dr. P. Albrecht
**dBASE II für den Schneider CPC**
**1985, 280 Seiten**
**Best.-Nr. MT 90188**
**ISBN 3-89090-188-3**
**DM 49,–/sFr. 45,10/öS 382,20**




Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0
Bestellungen im Ausland bitte an:
Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656
Österreich: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526
Ueberreuter Media Handels- und Verlagsges. mbH, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0
Irrtümer und Änderungen vorbehalten







Fragen Sie Ihren Buchhändler nach unserem kostenlosen Gesamtverzeichnis mit über 200 aktuellen Computerbüchern und Softwareprogrammen. Oder fordern Sie es direkt beim Verlag an!

gehören auch Bildschirmmodus und -Basisadresse, Interruptmodus, Window-Grenzen und die aktiven RAM- und ROM-Bänke. Und als kleinen Leckerbissen bietet das Multiface noch den Zugriff auf seinen eigenen 8 KByte großen RAM-Bereich. Dort kann der Maschinensprache-Programmierer in Zukunft seine Routinen unterbringen. Auch hier hat also der Herausforderer die Nase vorn.

Welches Modul empfehlen wir nun? Sieger nach Punkten ist nach wie vor eindeutig das neue Multiface Two. Das etwas unpraktische Gehäuse nimmt man dafür leicht in Kauf. Eine Warnung sei aber für all jene ausgesprochen, die ihren Computer mit zusätzlicher Peripherie wie beispielsweise einer Speichererweiterung ausgerüstet haben, oder dies jemals nachzuholen beabsichtigen. Können Sie das für sich ausschließen, ist das knapp 175 Mark teure (CPC-6128-Version zirka 180 Mark) Multiface Two der richtige Griff. Testen Sie am besten im Einzelfall die Verträglichkeit.

## Vorsicht beim Kauf

Der Mirage Imager ist zwar nicht so schnell und erlaubt nur beschränkte Eingriffe in die Programme, erfüllt aber ansonsten seine Aufgabe mit der gleichen Leichtigkeit. Auch gibt er sich erheblich verträglicher im Umgang mit fremden Peripheriegeräten. Das mag für manchen der ausschlaggebende Grund sein, lieber bei diesem Angebot für 179 Mark (198 Mark für den CPC 6128) zuzugreifen. Ganz auf den Genuß des Einsatzes eines der beiden Module müssen jedoch Besitzer von Diskettenlaufwerken verzichten, die nicht über den originalen Schneider-Controller angesteuert werden. Wie schon gesagt: Schade. (ja)

Mirage Microcomputers, Falkenweg 16, 5400 Koblenz 16, Telefon 0261/68734
Romantic Robot, Ben-Gurion-Ring 86, 6000 Frankfurt 56

# Was ist bloß mit den Programmen los?

**Immer wieder erreichen uns Anfragen zu den veröffentlichten Programmen. Deshalb hier ein paar grundsätzliche Worte und Tips.**

Beginnen wir mit den gedruckten Listings. Trotz Explora kommt es immer wieder vor, daß das eben eingegebene Programm nicht korrekt läuft. Aber statt zu verzweifeln, überprüfen Sie bitte Ihre Eingabe nochmals genau. Selbst mit einem Prüfsummer ist die Eingabe nicht hundertprozentig sicher, denn manche Fehler des Benutzers kann auch ein solches Programm nicht abfangen. Dazu zählt beispielsweise die vergessene Eingabe einer kompletten Zeile. Aber auch der genauso banale wie häufige Fehler der Nichtbeachtung von Prüfsummen während der Eingabe ist tükkisch. Dabei sind die beiden geschilderten Fehler auch im nachhinein recht einfach zu erkennen. Ein erneuter Durchlauf mit Explora wirkt oft Wunder. Haben Sie keine Angst, Sie müssen selbstverständlich das Programm nicht nochmals komplett eingeben. Nein, Sie nutzen den AUTO-Befehl Ihres Computers, um das Programm im Arbeitsspeicher zu kontrollieren. Nur Besitzer eines CPC 464 können sich so nicht direkt helfen. Sie müssen zunächst das Listing »AUTO-Plus« aus dieser Sonderausgabe eingeben, um die gleichen Voraussetzungen zu erfüllen wie die Kollegen mit den CPCs 664 und 6128. Laden Sie zunächst das fehlerhafte Programm (CPC 464-Besitzer starten natürlich davor noch »AUTO-Plus«). Mit dem Befehl »AUTO zeilennummer,,schrittweite« wählen Sie die Nummer der ersten zu kontrollierenden Zeile und die Schrittweite der Numerierung. Liegt ein Listing vor, dessen Numerierung nicht in gleichmäßigen Schritten erfolgt, setzen Sie als letzten Parameter eine 1 ein. Der Computer zeigt Ihnen darauf jede Zeile auf dem Bildschirm. Überprüfen Sie die Übereinstimmung der Zeilennummern auf dem Monitor und im Listing. Der Inhalt der Zeilen interessiert in diesem Moment nicht. Nach der Kontrolle der Zeilennummer drücken Sie <ENTER>, woraufhin Explora seine Prüfsumme auf dem Bildschirm ausgibt und der Editor schon die nächste Zeile zeigt. Jetzt läßt sich die Prüfsumme auf Richtigkeit vergleichen. Einfach per Druck auf die ENTER-Taste sind Sie so ruck-zuck durchs Programm und haben alle Fehler beseitigt. Wenn Sie zu den Lesern gehören, die öfter Listings von Basic-Ladern (DATA-Zeilen mit Maschinencode) abtippen, sei Ihnen das Programm »CPC« – ebenfalls in dieser Ausgabe – wärmstens empfohlen. Damit sparen Sie sich einen Großteil der Tipparbeit.

## Leserservice gutgemeint

Auch unser Angebot der Leserservice-Disketten und -Kassetten ist häufig Gegenstand von Anfragen. Was uns dazu interessiert, ist, in welcher Form Sie sich die Dateinamen wünschen. Manche Leser bevorzugen abstrakte Dateinamen, die aus der Ausgaben- und Seitennummer bestehen. Andere wieder lehnen diese Namensgebung ab und sind erfreut über die derzeitige Form mit sinnvollen Bezeichnungen dessen, was die Programme bewirken. Wir jedenfalls glauben, daß man mit einem Namen wie beispielsweise »HAUSHALT.BAS« eher ein Programm zur Berechnung von Haushaltskosten assoziiert, als wenn dieselbe Datei »SH13.112« heißt. Da uns aber nur vereinzelte Meinungsäußerungen vorliegen, die keine Rückschlüsse auf die Wünsche der Mehrzahl unserer Leser erlauben, bitten wir Sie, sich mit einer kleinen Stellungnahme auf einer Postkarte bei uns zu melden. In einer Sache haben wir aber zu einer endgültigen Form gefunden. Die Datei, die Erklärungen des Inhalts der Leserservice-Datenträger enthält, heißt seit einiger Zeit und in Zukunft immer »README.BAS« und ist ein Basic-Programm, das Sie normal mit »RUN "README"« starten. So erhalten Sie zu jeder Datei die Information, was sie enthält und auf welcher Seite der jeweiligen Ausgabe gedruckt zu finden ist. Auf einigen früheren Datenträgern hieß diese Datei »LISTME.BAS«. (ja)

# Die Klassiker-Kollektion

**Auch wenn Sie kein ausgesprochener Spiele-Freak sind, sollten Sie trotzdem weiterlesen. Hier stellen wir Ihnen die Programme vor, die in keiner Software-Sammlung fehlen sollten.**

In den letzten Jahren haben wir regelrechte Hundertschaften von Schneider-Spielen getestet. Wenn man im Rückblick eine Liste derjenigen Programme aufstellt, die am meisten gefesselt haben, wird dies eine hochkarätige Reihe. Genau das wollen wir in diesem Sonderheft einmal praktizieren: Wir verraten Ihnen die Schneider-Spiele, die uns heute noch am besten gefallen und auch bei Leuten, die nur »nebenher« spielen, Begeisterung hervorrufen.

## Die Effekt-Orgie

Die Auswahl ist gezwungenermaßen subjektiv, aber keineswegs willkürlich. Sie entstand in einer Diskussion aller Redakteure, die sich für CPC-Spiele interessieren. Ein Grafik-Adventure ist leider nicht vertreten, da uns bisher noch kein Spiel dieser Sparte aufgefallen ist, das den Schneider CPC besonders gut ausnutzt. Aber es besteht noch Hoffnung: Das Abenteuerspiel »The Pawn«, das bereits Lobeshymnen für die Atari-ST- und C64-Versionen erntete, erscheint bald auf Diskette für Schneider. Wer sehr gut Englisch kann und vor vielen Texten nicht zurückschreckt, dem seien generell die Adventures von Infocom empfohlen. (Unser aktueller Tip: das witzige »Leather Godesses of Phobos«, für das man allerdings gute Englischkenntnisse unbedingt benötigt.) Aber jetzt geht's los mit unserer Wertung.

Umsetzungen von Spielhallen-Automaten sind in der Software-Branche gang und gäbe. Leider verlieren diese Adaptionen bei den Heimcomputer-Umsetzungen an Qualität. Das liegt zum einen an den exzellenten Grafik- und Sound-Fähigkeiten der Spielautomaten, zum anderen am ganz besonderen Flair, das in einer Spielhalle »rüberkommt«.

Die Schneider-Version des »Tempest«-Automaten ist eine der rühmlichen Ausnahmen. Die Vektorgrafik des Originals wurde sehr flott auf dem CPC programmiert und die Sound-Effekte lassen so manches Amiga-Spiel arm aussehen. Man muß den Schneider für den vollen Klanggenuß allerdings an eine Stereoanlage anschließen, doch dann verwandelt sich das heimische Wohnzimmer regelrecht in eine Spielhölle.

Tempest ist ein simples Ballerspiel, bei dem es nur um gute Reflexe geht. Die reizvolle Vektorgrafik, die 99 verschiedenen Bilder und die imposante Explosions-Geräuschkulisse sorgen aber für ungeheuren Spielspaß. Wer ein unkompliziertes, technisch toll gemachtes Ballerspiel sucht, sollte hier unbedingt zuschlagen (Kassette 35 Mark, Diskette 49 Mark).

## Der Dauerbrenner

Wer träumt nicht einmal davon, von Galaxis zu Galaxis zu fliegen, fremde Planeten zu besuchen, Handel zu treiben und Raumpiraten zu jagen? 1985 erschien die Schneider-Version eines Kultspiels, das solch eine kosmische Odyssee möglich macht. »Elite« nennt sich das gute Stück, das auch nach Monaten noch eine Menge Spielspaß bietet. Dadurch, daß man einen Spielstand speichern kann, läßt sich die Weltraumreise beim nächsten Mal wieder an der gleichen Stelle fortsetzen.

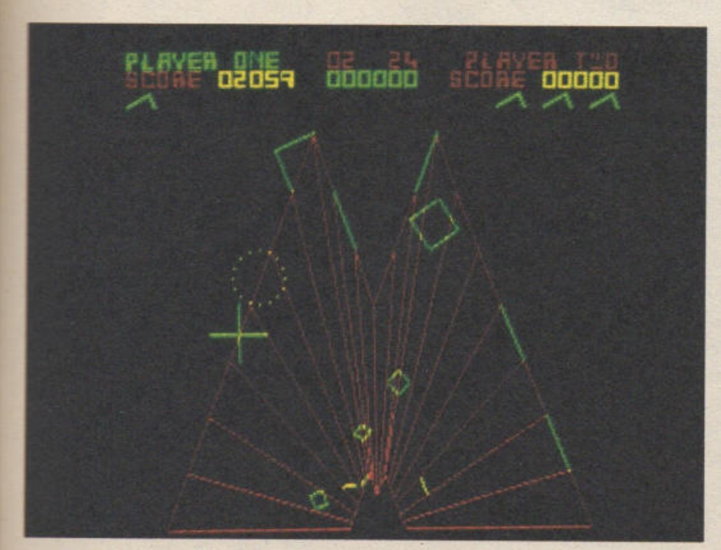
Elite ist eine Mischung aus Flugsimulation, Handels- und Actionspiel, die mit ihren 3D-Vektorgrafiken für Aufsehen sorgte. Außerdem war Elite eines der ersten Spiele, bei denen sowohl Anleitung als auch Bildschirmtexte ins Deutsche übersetzt wurden. Dafür muß man einige sprachliche Holprigkeiten (wurde wohl 1:1 mit dem Wörterbuch übersetzt) verzeihen.

Obwohl auch etwas geschossen wird, paßt Elite nicht in die Action-Schublade. Das faszinierende Spielprinzip und die strategischen Elemente beim Handel auf den über 2000 Planeten machen es zu einem anspruchsvollen Vergnügen, das kaum jemanden kalt läßt. Nicht nur für Science-fiction-Fans dringend empfohlen (Kassette 59 Mark, Diskette 69 Mark).

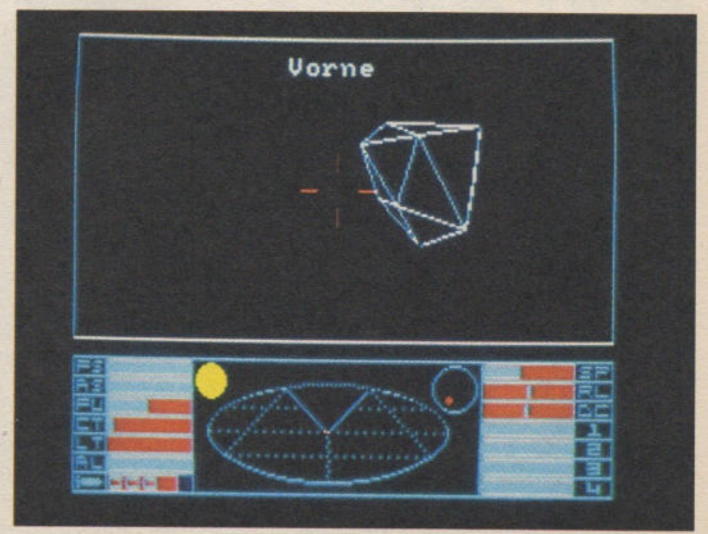
## Der Bomben-Klau

In eine ganz andere Kerbe schlägt unser dritter Spiele-Tip. »Bomb Jack« ist nicht sonderlich kompliziert, aber es wird auch nicht auf irgendwas geschossen. Das Spiel ist ebenso herzallerliebst wie fesselnd und damit auch für Kinder gut geeignet.

Der Spieler muß eigentlich »nur« brennende Bomben abräumen, die auf Plattformen plaziert sind. Dabei behindern ihn aber Gegner, die er nicht berühren darf. Um Höchstpunktzahlen zu erreichen, muß man möglichst alle

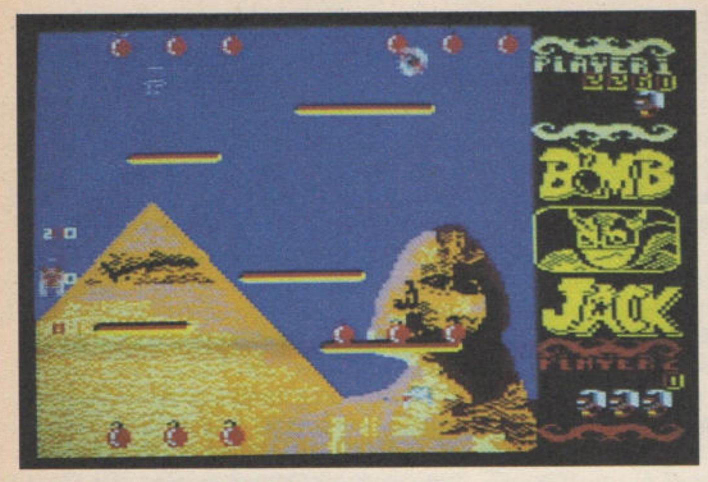



Grafik- und Sound-Orgie für Actionfans: Tempest

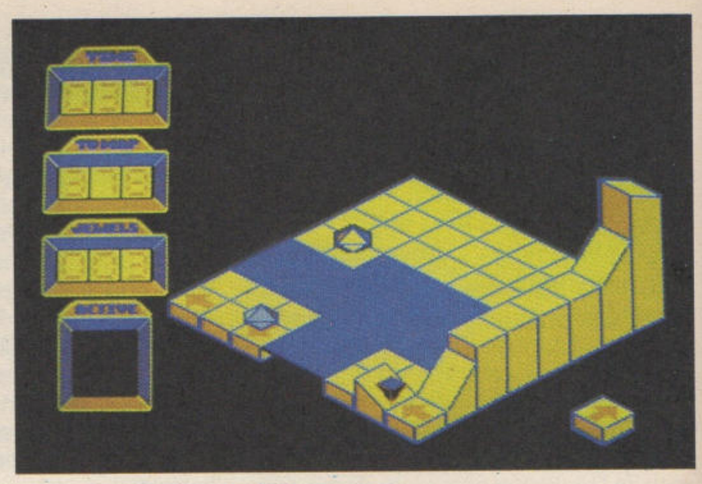



Weltraum-Trip für Erdenbürger: Elite





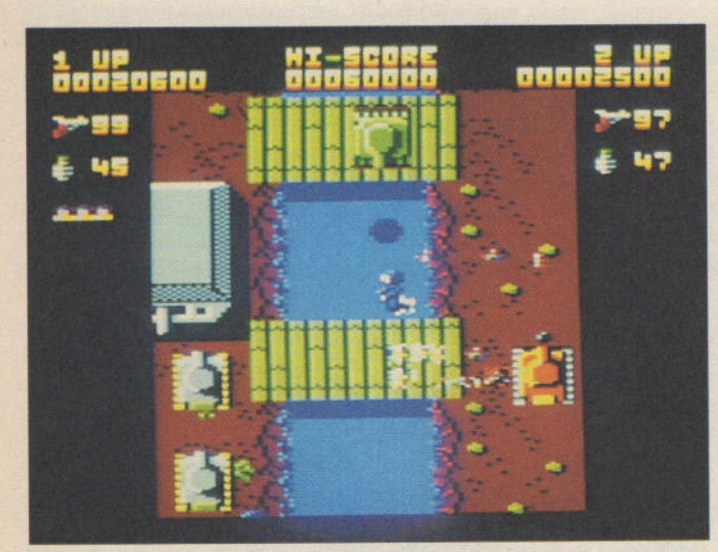

Bombige Punktejagd: Bomb Jack



Diamanten-Fieber: Spindizzy

Bomben in einer bestimmten Reihenfolge erwischen, was immer einen enormen Punkte-Bonus einbringt.

Da sich der Aufbau der Plattformen von Bild zu Bild ändert, entscheidet jeweils eine andere Strategie, wie optimal Sie jeden Level abräumen. Die Punktejagd bei Bomb Jack gehört zum Motivierendsten, was je auf einem CPC gelaufen ist. Das Programm ist nicht umsonst die Umsetzung eines Spielautomaten, der ja durch fesselndes Spielprinzip möglichst viele Mark-Stücke schlucken soll. Wenn jeder Redakteur eine Mark an mich gezahlt hätte, als das Programm in einer heißen Phase ständig bei uns gespielt wurde, könnte ich mich jetzt für ein Weilchen zur Ruhe setzen (Kassette 35 Mark, Diskette 49 Mark).




Ballerzeit zu zweit: Ikari Warriors

## Das flotte Doppel

Bleiben wir doch gleich bei Umsetzungen von Spielautomaten. Unter dem Motto »Grausam, aber gut« könnte man »Ikari Warriors« präsentieren. Das Programm ist der vorläufige Höhepunkt der kriegerischen Welle.

Wie bei anderen Programmen dieses Genres kämpfen Sie sich in der Gestalt eines knüppelharten Soldaten durch den Dschungel, um dort feindliche Gebäude zu vernichten und den gegnerischen Angreifern Saures zu geben. Doch nach dem Motto »Warum alleine schießen, wenn's zu zweit mehr Freude macht« können zwei Spieler gleichzeitig antreten und im Team alles über den Haufen schießen, was kreucht und fleucht. Jeder Spieler hat sein eigenes Punktekonto, so daß einem spannenden Simultan-Wettkampf nichts im Wege steht.

Wenn man sich nicht gerade an der recht rauhen Handlung stört, macht das Programm (auch alleine) einen Heidenspaß. Hier darf man nach Herzenslust alles demolieren, was einem so in den Weg kommt. Wenn Sie sich jetzt noch einen herumstehenden Panzer schnappen, geht es richtig rund. Jetzt kann man fast ohne Rücksicht auf Verluste durch die Gegend rumpeln und gepflegt die Landschaft in Einzelteile zerlegen. Je weiter man vordringt, desto mehr neue Gegner tauchen auf. Mal muß man einen Fluß durchschwimmen und sich vor Heckenschützen hüten, dann erfolgt ein Hubschrauber-Angriff aus der Luft.

Ikari Warriors ist kein sonderlich friedliebendes Spiel, aber ebenso destruktiv wie unerhört unterhaltsam. Wer dieses Programm hat, kann eigentlich alle anderen »Dschungel-Päng-Päng«-Spiele für den Schneider vergessen (Kassette 39 Mark, Diskette 59 Mark).

Der bis dato recht unbekannte Programmierer Paul Shirley sorgte Anfang 1986 für einen der größten Knaller der Software-Branche. Er präsentierte nämlich ein Schneider-Spiel, das hervorragende 3D-Grafik, ungeheure Komplexität (über 300 Bilder) und ein raffiniertes Spielprinzip vereinte.

»Spindizzy« nennt sich dieses Prachtstück und gehört zu den absoluten Klassikern für den CPC. Hier geht es um die Erforschung einer neuen Welt, die links hinter der 17. Dimension entdeckt wurde. Dieser merkwürdige Ort besteht aus Rampen, Aufzügen, Trampolinen und Einbahnstraßen, aber auch aus ein paar gefährlichen Einheimischen.

## Die Wunderwelt

Sie steuern ein Fahrzeug mit dem Spitznamen Gerald, mit dem Sie diese Wunderwelt erforschen und kartographieren sollen. Diese Expedition wird zu einem Rennen gegen die Zeit, die ständig abläuft. Durch das Aufsammeln von Diamanten, die freundlicherweise herumliegen, kann man aber wieder wertvolle Sekunden dazugewinnen.

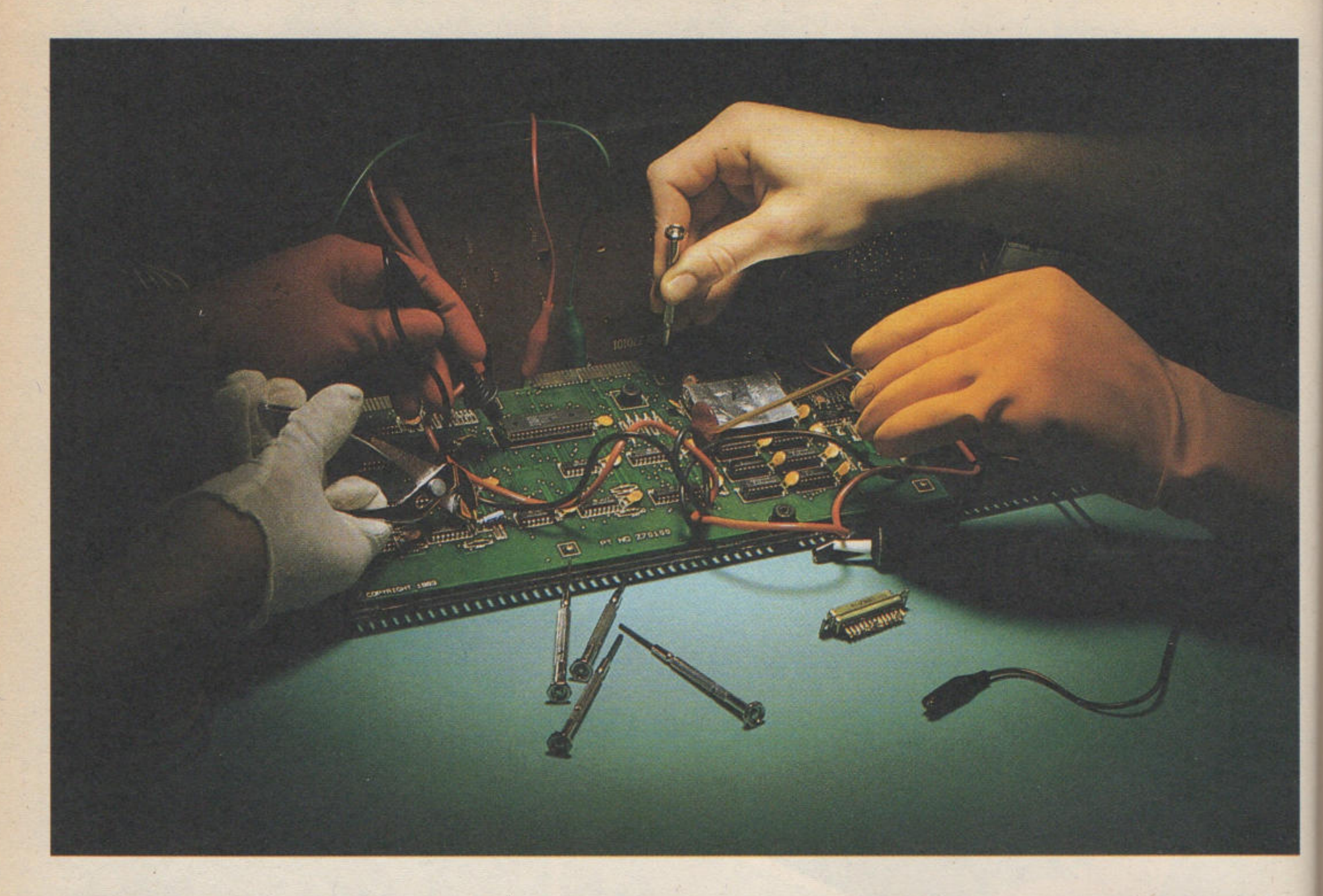
Spindizzy ist neben Bomb Jack das mit Abstand beste Geschicklichkeitsspiel für den Schneider und besticht neben der famosen Grafik durch das Spielprinzip. Um bei einigen Bildern weiterzukommen, muß man auch Mal ein logisches Rätsel lösen. Ein garantierter Spielgenuß mit Langzeit-Wirkung (Kassette 35 Mark, Diskette 49 Mark).

Soweit der Blick in unsere Schneider-Ehrengalerie. Wenn Sie jetzt auf den Geschmack gekommen sind und sich über brandneue Spiele für Ihren CPC informieren wollen, empfehlen wir unsere monatlich erscheinende Stammzeitschrift Happy-Computer. In jeder Ausgabe gibt es einen extra Spiele-Teil, der Sie über die aktuellen Neuheiten auf dem laufenden hält. (hl)

Nähere Informationen zu den Spielen erhalten Sie bei folgenden Distributoren:
Infocom-Adventures, Tempest, Spindizzy: Activison Deutschland, Postfach 75 06 80, 2000 Hamburg 76
Elite: Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2
Bomb Jack, Ikari Warriors: Peter West Records, Am Heerdter Hof 15, 4000 Düsseldorf 11



Markt&Technik
1/87
68000er
DAS MAGAZIN DER NEUEN COMPUTER-GENERATION
JANUAR 1987
DM 7,-
Atari ST, Amiga und QL
Starthilfe
• Umstieg ohne Verluste: Zubehör weiter nutzen
• Test: Die passenden Drucker
Jack Tramiel: Meine Pläne für 1987
Programmieren
• Alle Sprachen auf einen Blick
• Große Basic-Vergleichstabelle
• 68000er-Referenzkarte: GfA-Basic
• Test: Multitasking-Betriebssystem OS-9
• Anspruchsvolle Listings
Erstausgabe
AUFBRUCH IN EINE NEUE DIMENSION
mit »68000er«, dem Magazin der neuen Computer-Generation. Die Erstausgabe 1/87 startet am 15. 12. 86 mit den Themen:
Geheimnisse gelüftet:
Kurse für den souveränen Umgang mit dem Amiga.
Komfort mit Basic:
Professionelles Malprogramm für den Atari ST.
Selbstbau:
RAM-Erweiterung und Digitizer für den Amiga.
Die neue Spielegeneration:
Flugsimulation für Atari ST und Amiga.
Ready for take off am
15. DEZEMBER 1986
... und danach jeden Monat neu:
»68000er«, Ihre hot-line zur Spitzentechnologie von Atari ST, Amiga, Macintosh und Sinclair QL.
GUTSCHEIN
FÜR EIN KOSTENLOSES PROBEEXEMPLAR DES »68000er«-MAGAZINS
JA, ich möchte »68000er«, das Magazin der neuen Computer-Generation, kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »68000er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte es dann regelmäßig frei Haus per Post. Außerdem nutze ich den Abonnement-Preisvorteil von 8% und bezahle pro Jahr nur 77,– DM statt 84,– DM im Einzelverkauf.
Vorname
Name
Straße
PLZ, Ort
Datum 1. Unterschrift
Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.
Datum 2. Unterschrift
Gutschein ausfüllen und absenden an: Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar
HCS187
WWW.HOMECOMPUTERWORLD.COM



# CPC auf dem Operationstisch

**Die Schneider-Computer arbeiten in der Regel sehr zuverlässig, doch für die Ewigkeit sind auch sie nicht konstruiert. Sollte Ihr Gerät einmal streiken, läßt sich der Schaden oft mit wenig Aufwand selbst beheben.**

Der Ausfall des Computers ist der Alptraum jedes Computerbesitzers. Und das passiert natürlich zum ungünstigsten Zeitpunkt, wenn das neue Spiel gerade geladen oder ein dringender Brief fällig ist.

Sie können natürlich Ihr Gerät in den Karton packen, beim Händler vorbeibringen, oft mehrere Wochen warten und anschließend einen stolzen Betrag für die Reparatur entrichten. Zurück bleibt dann aber in der Regel ein bitterer Nachgeschmack. War die Reparatur wirklich ihr Geld wert? Hätte man den Schaden eventuell nicht selbst beheben können?

Wir wollen Ihnen zeigen, wie Sie Fehler im Computersystem ohne großen Aufwand finden und selbst beheben. So müssen Sie nicht wochenlang auf Ihr Gerät verzichten und zum anderen schont es den Geldbeutel.

Um eine Störung zu beseitigen, muß man erst einmal wissen, wo sie steckt. Diese simple Weisheit stellt den Computerbesitzer vor große Probleme, denn die Fehlersuche ist meist weitaus aufwendiger als die Reparatur selbst.

Wichtig ist zuerst einmal festzustellen, ob der Fehler reproduzierbar (das heißt beliebig oft wiederholbar) ist. Wenn eine Störung nur zeitweise und völlig unmotiviert auftritt, handelt es sich entweder um einen Softwarefehler (den wir hier nicht behandeln wollen) oder um einen Wackelkontakt. Um diesen Mangel zu beheben, überprüfen Sie zuerst alle äußeren Steckverbindungen des Computers. Findet sich hier kein Fehler, müssen Sie sich an das Innere Ihres Gerätes wagen. Aber keine Angst, bei Ihrem Schneider ist alles schön verschraubt und somit leicht zerleg- und wieder zusammensetzbar.

## Erste Hilfe

Bevor Sie mit dem Zerlegen des Gerätes beginnen, ziehen Sie unbedingt den Netzstecker aus der Steckdose! Um das Gehäuse des CPCs zu öffnen, lösen Sie zuerst die sechs (CPC 464/664) beziehungsweise sieben (CPC 6128) Schrauben an der Unterseite des Gerätes. Beim CPC 664 und 6128 ist zusätzlich das Entfernen der beiden Schrauben an der rechten Gehäuseseite zum Diskettenlaufwerk nötig. Jetzt kann das Gehäuseoberteil mit Tastatur vorsichtig abgehoben werden. Ober- und Unterteil hängen jedoch immer noch an Kabeln aneinander.

Grundsätzlich ist die Tastatur im Gehäuseoberteil über ein Flachkabel mit der Platine im Computerunterteil verbunden. Beim CPC 664 kommen noch der Anschluß für den Lautsprecher und beim CPC 6128 der Anschluß für Lautsprecher und Betriebsanzeige (Leuchtdiode) hinzu. Diese Verbindungen müssen Sie ebenfalls vorsichtig lösen, um den Computer frei bewegen zu können. Bei alten Versionen des CPC 464 kann zusätzlich das Kühlblech über dem Gate Array entfernt werden. Um beim CPC 6128, der nach den neuen Bestimmungen der Bundespost hergestellt wird, an die Platine zu gelangen, ist noch die Abschirmung, die mit sechs Schrauben auf der Platine befestigt ist, abzunehmen.

Nun liegt in allen drei Fällen die Platine des CPCs frei. Bild 1 zeigt die Platine

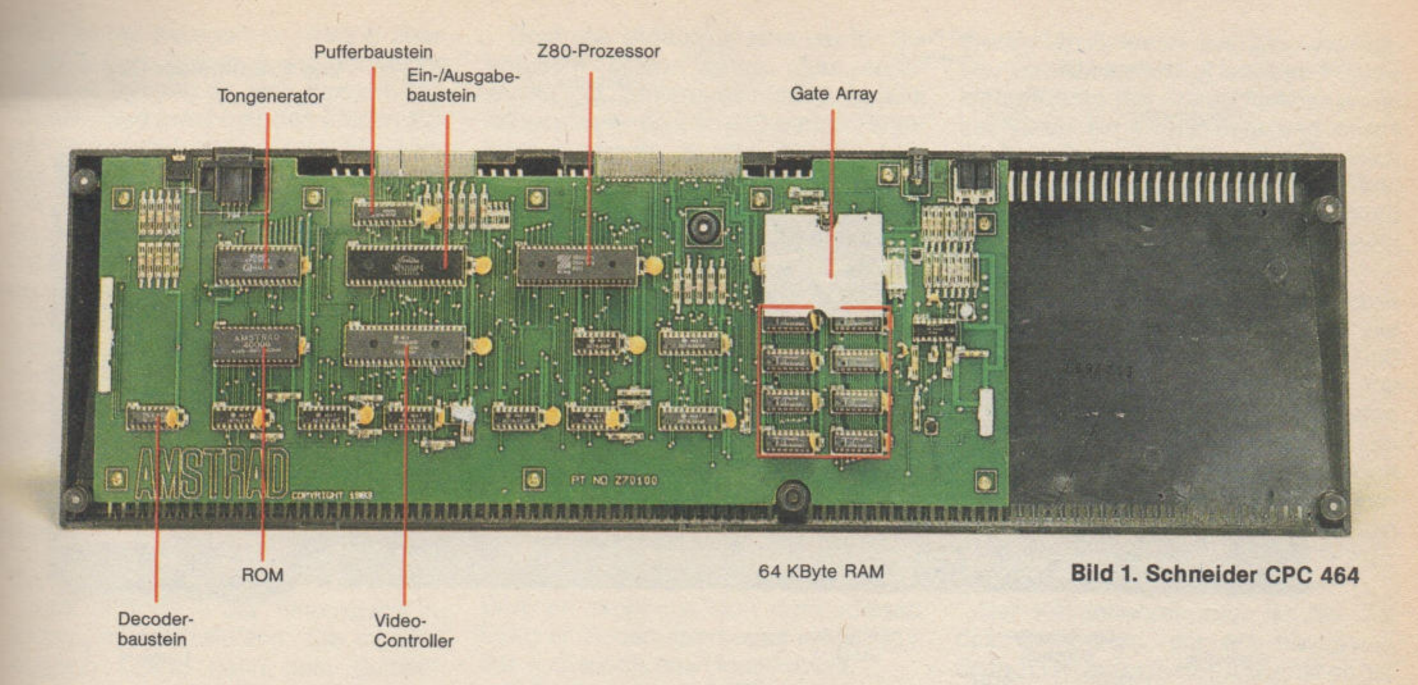



**Bild 1. Schneider CPC 464**

des CPC 464, Bild 2 die Platine des CPC 664 und Bild 3 die des CPC 6128 mit Benennung der wichtigsten Bausteine. Um einen Wackelkontakt zu beheben, überprüfen Sie alle Bausteine im Computer auf festen Sitz und testen die Steckverbindung zum Kassettenrecorder (CPC 464) beziehungsweise Diskettenlaufwerk (CPC 664/6128). Ist der Wackelkontakt immer noch vorhanden, so macht ihm spätestens das sorgfältige Zusammenbauen des Gerätes den Garaus. Dann werden nämlich alle in Frage kommenden Verbindungen neu geschlossen und lose sitzende Stecker oder Buchsen können erkannt und ersetzt werden. Erst wenn Ihr Schneider wieder komplett verschraubt vor Ihnen steht, dürfen Sie das Gerät an das Netz anschließen und einschalten.

Tritt ein Fehler im System regelmäßig nach längerer Computerbenutzung auf, handelt es sich oft um die Auswirkungen übermäßiger Erhitzung. In diesem Fall kontrollieren Sie zuerst, ob die Entlüftungsschlitze von Computer und Monitor frei liegen. Wenn sich hier dicke Staubschichten angesammelt haben oder diverse Papiere stapeln, kann die erwärmte Luft aus dem Gerät nicht entweichen und es entsteht ein Hitzestau.

## ICs mögen's kühl

Schafft das Freilegen der Lüftungsschlitze keine Abhilfe, dann geht es Ihrem CPC wieder an den Kragen und sämtliche ICs werden auf Erwärmung überprüft. Ist ein einzelner Baustein mehr als handwarm, können Sie ihn also nur kurz berühren, ohne sich die Finger zu verbrennen, ist dieser IC defekt und muß ersetzt werden. Dies sollten Sie umgehend in Angriff nehmen, weil ein defekter Baustein in seiner Hardware-Umgebung weitere Ausfälle bewirkt. Wollen Sie Ihre Diagnose bombensicher machen, kaufen Sie sich im Elektronikladen ein Kältespray und kühlen Sie den Chip damit ab. Verschwindet der Fehler daraufhin für einige Zeit, ist die Ursache eindeutig der Chip.

Ein »IC ersetzen« hört sich zwar einfach an, ist aber eine Wissenschaft für sich. Leider sind die meisten Bausteine in den CPCs nicht gesockelt, so daß sie beim Ersetzen ausgelötet werden müssen. Anschließend wird erst ein Sockel eingelötet und dann der neue IC einge-

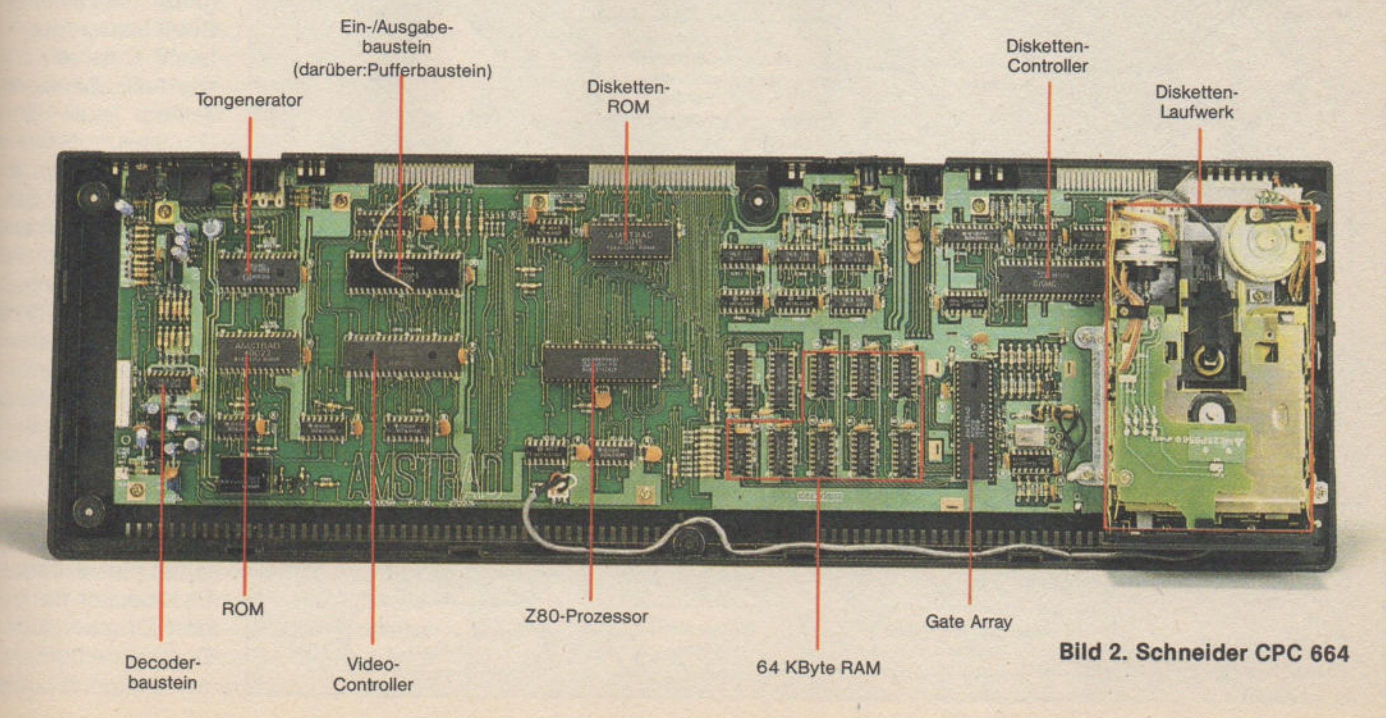



**Bild 2. Schneider CPC 664**

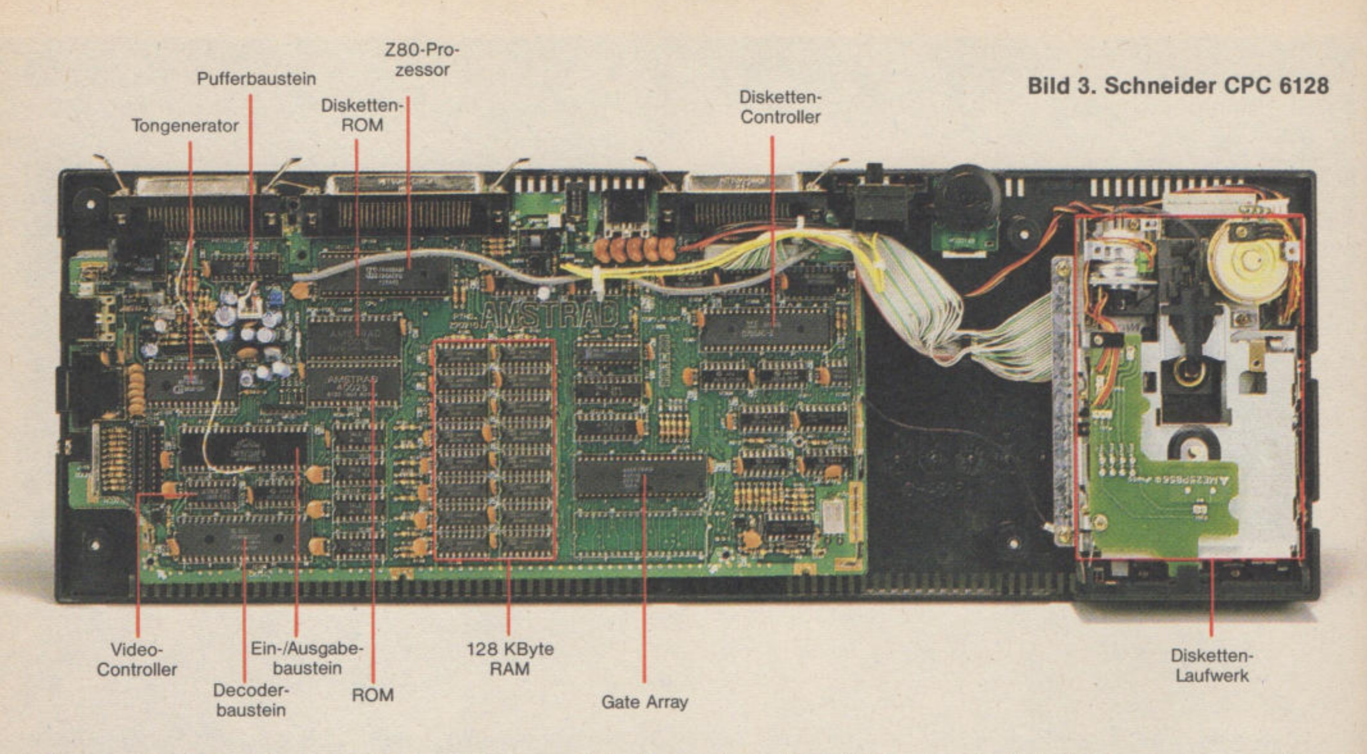



**Bild 3. Schneider CPC 6128**

setzt. Ein erneuter Bausteinwechsel ist jetzt unproblematisch.

Zum Auslöten sind jedoch handfeste Lötkenntnisse Voraussetzung. Wer sich eine solche Reparatur selbst nicht zutraut, findet sicher in seinem Bekanntenkreis jemanden, der ihm weiterhilft.

Des weiteren erfordert diese Prozedur eine gewisse Ausrüstung. Hierzu zählt unbedingt ein Lötkolben mit einer maximalen Leistung von 25 W und eine Entlötpumpe oder -litze beziehungsweise ein Entlötkolben. Beim Auslöten ist die Lötzeit relativ unwichtig (bei extrem langen Lötzeiten können sich allerdings in näherer Umgebung die Leiterbahnen von der Platine lösen). Wenn Sie dagegen das IC wieder verwenden wollen, darf eine Lötzeit von drei bis fünf Sekunden pro Pin (IC-Beinchen) nicht überschritten werden. Daß beim Ein- und Ausbau von Bausteinen im Computer der Netzstecker gezogen sein muß, ist selbstverständlich.

Ein weiterer Grund, warum ein CPC 6128 nach einer gewissen Zeit ausfallen kann, besteht darin, daß ein Speicherbaustein der zweiten RAM-Bank (zweite Seite im Arbeitsspeicher) ausgefallen ist. Diese RAM-Bank wird nur von CP/M Plus, von bestimmten Maschinensprache-Programmen und von anderen sehr langen Programmen aufgerufen. Falls Ihr Gerät also in einem dieser Fälle beharrlich aussteigt, handelt es sich um den erwähnten Defekt eines Speicherbausteins. Diese Art von Störung wird im letzten Abschnitt dieses Beitrages ausführlich behandelt.

Gewöhnlich tritt eine Störung im Computer bei einer ganz bestimmten Befehlssequenz (zum Beispiel bei der Ausgabe eines Textes auf den Drucker) auf. Oder die Fehlfunktion ist schon beim Einschalten vorhanden, das heißt, der Computer verweigert die Annahme jeglicher Eingaben über die Tastatur, und auf dem Bildschirm präsentiert sich ein wirres Chaos.

Im ersten Fall läßt sich der Fehler leicht lokalisieren. Es kann sich nur um den Drucker oder Druckertreiber handeln. Im zweiten

Tonstörung
externer Verstärker angeschlossen?
nein
ja
Verstärker abklemmen und internen Lautsprecher ganz aufdrehen
Ton?
nein
ja
schwaches Geräusch?
nein
ja
1,5-V-Batterie über Draht direkt an den Lautsprecher des CPC halten
Geräusch?
nein
ja
Andere Tonquelle (z.B. Radio) an externen Verstärker anschließen
Tongeneratorbaustein auswechseln
Ton?
nein
ja
4,5-V-Batterie über Draht direkt an den Lautsprecher des externen Verstärkers halten
Internen Verstärker reparieren
nein
Ton?
ja
Kabel zum Verstärker ersetzen
Geräusch?
nein
ja
Tongenerator war defekt
Externen Verstärker reparieren
Lautsprecher ersetzen
Internen Lautsprecher ersetzen

**Bild 4. Flußdiagramm »Tonstörung«**

Fall gibt es dagegen zu Anfang überhaupt keinen Anhaltspunkt, um welche Art von Störung es sich handelt. Der Unterschied zwischen den beiden Fällen besteht darin, daß im ersten Beispiel das Betriebssystem des Computers noch arbeitet und nur ein Teil der Funktionen des Gerätes gestört ist. Im zweiten Beispiel handelt es sich dagegen um einen Totalausfall.

Entsprechend ihrem Schwierigkeitsgrad kommen im folgenden zuerst Störungen bei funktionsfähigem Betriebssystem und darauf die Totalausfälle zur Sprache.

## Fehler im System

**Tonstörungen:**

Eine Störung in der Tonausgabe zu finden, ist sehr einfach. Sie müssen lediglich streng nach dem Schema unseres Reparatur-Flußdiagrammes in Bild 4 vorgehen, und zum Schluß die Anweisung, die in dem für Sie infragekommenden Feld steht, befolgen. Die Reparatur des Verstärkers führt am besten entweder ein fachkundiger Bastler oder eine Reparaturwerkstatt durch.

Zum Wechseln des Tongenerators ist zu bemerken, daß Sie selbstverständlich zum Testen den Baustein eines anderen CPC-Besitzers mit funktionierender Tonausgabe verwenden können. Wenn es mit der Tonausgabe dann immer noch nicht klappt, haben Sie wenigstens das Geld für einen zweiten Baustein gespart. Außerdem müssen Sie die weiter oben gegebenen Hinweise zum Ein- und Ausbau von ICs unbedingt beachten.

**Bildstörungen:**

Das Flußdiagramm zur Störungssuche bei der Bildausgabe zeigt Bild 5. Ein erneutes Einschalten des Computers und die Eingabe des SOUND-Befehls ist deshalb sinnvoll, weil das gesamte Computersystem abgestürzt sein könnte. Dies hätte unter Umständen ebenfalls eine fehlerhafte Bildausgabe zur Folge. Mit dem SOUND-Befehl wird nun überprüft, ob das Betriebssystem noch arbeitet und den Ton ausgibt. Ist dies nicht der Fall, so liegt der Fehler an einer anderen Stelle.

Wenn sowohl das Auswechseln des Gate Arrays, als auch das Auslöten und Ersetzen des Videocontrollers (hier gilt das gleiche wie beim Ein- und Ausbau des Tongenerators) nicht zum gewünschten Erfolg führt, muß der Fehler am Monitor liegen. Da der Monitor jedoch intern mit Spannungen von bis zu 16 Kilovolt (Grünmonitor) beziehungsweise 24 Kilovolt (Farbmonitor) arbeitet, muß man das Gerät unbedingt zum Fachhändler bringen.

Doch die Störungssuche hat für Sie trotzdem einen Vorteil. Zum einen wurde die Fehlerquelle damit bereits näher eingekreist, so daß bei der Reparatur geringere Kosten für die Störungssuche anfallen. Zum anderen können Sie den Fachhändler von Ihren technischen Kenntnissen überzeugen, wenn Sie ihm Ihre bisherigen Ergebnisse bei der Fehlersuche schildern. Die Gefahr, bei der Kalkulation der Reparaturrechnung übervorteilt zu werden, wird dadurch geringer.

**Druckerstörungen:**

Auch bei der Störungssuche infolge fehlerhafte Druckerausgabe wird nach einem festen Schema vorgegangen, das Sie in Bild 6 finden. Bedingt durch das komplizierte Zusammenspiel von Mechanik und Elektronik in einem Drucker, empfiehlt sich dem völligen Laien an Reparaturarbeiten jedoch nur der Austausch des Druckkopfes.

Beim Austausch des Pufferbausteins im CPC sollten wieder die weiter oben erwähnten Hinweise zum Ein- und Ausbau von ICs beherzigt werden. Für die Überprüfung der Treiberfunktionen ist zusätzlich ein Oszilloskop und etwas Erfahrung im Verfolgen von Signalen und Leiterbahnen erforderlich. Die Treiberbausteine sind in der Regel nicht gesockelt und müssen ebenfalls durch Auslöten ersetzt werden.

Das Auswechseln der Schrittmotoren ist oft sehr aufwendig, weil sie die Schnittstelle zwischen Elektronik und Mechanik darstellen und dadurch mit beiden Komponenten direkt verbunden sind. Das Ablöten der elektrischen Leitungen gelingt noch relativ leicht, doch der mechanische Aus- und Einbau verlangt schon einiges an handwerklichem Geschick.

**Recorderstörungen:**

Der Kassettenrecorder ist der Teil des Computers, der am häufigsten Grund zum Ärger bietet. Dabei ist es gar nicht so schwer, hier für Abhilfe zu sorgen. Am häufigsten ist die falsche Ein-

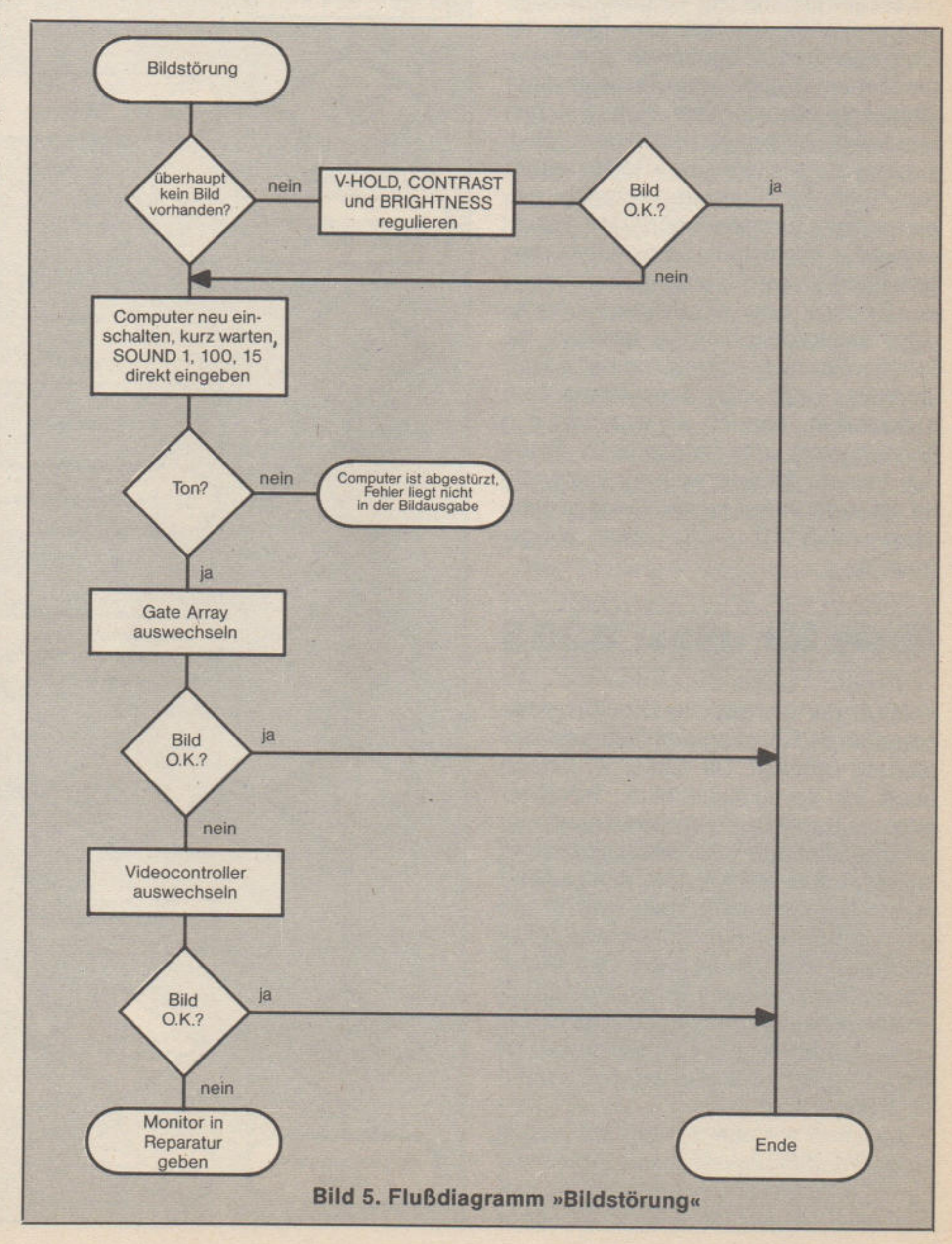


Bild 5. Flußdiagramm »Bildstörung«

stellung des Tonkopfes die Ursache des Ärgers. Die Meldung »load error« hat sicherlich jeder CPC-Besitzer schon einmal auf dem Bildschirm entdeckt. Das Flußdiagramm in Bild 7 geht auf diese und weitere Störungsmöglichkeiten ein.

Der Austausch des Relais oder des Motors bleibt wiederum nur erfahrenen Bastlern vorbehalten. Die Reparatur des Verstärkers im Recorder kann nur ein Fachmann durchführen.

**Diskettenstörungen:**

Fehler bei der Diskettenverwaltung des CPCs zu finden und zu beheben, fällt dem Laien sehr schwer. Der Fehler liegt entweder im Disketten-Controller oder in der Mechanik des Laufwerks. Doch Wartungs- und Reparaturarbeiten auf diesem Gebiet können mehr Schaden anrichten (zum Beispiel die Spur verstellen) als helfen. Deshalb sollte auch hier nur der Fachhändler tätig werden.

**Tastaturstörungen:**

Daß die Tastatur des CPC bei der Eingabe streikt, ist fast ein Ding der Unmöglichkeit. Abgesehen von gelegentlichem Prellen (eine Taste wird nur einmal gedrückt, aber das Zeichen erscheint mehrmals auf dem Bildschirm) dürften hier keine Fehler auftreten. Sollte Ihr Computer trotzdem einmal jegliche Eingabe ignorieren, so sitzt entweder der Stecker des Kabels von der Tastatur zur Platine nicht richtig in der Buchse, oder ein Baustein, der an der Tastaturdekodierung beteiligt ist, funktioniert nicht korrekt. Ist neben der Tastatur auch der Joystickanschluß ausgefallen, handelt es sich um den Tongenerator, der nebenbei für Tastatur- und Joystickverwaltung zuständig ist. Im anderen Fall hat der Decoderbaustein 74LS145 sein Leben ausgehaucht.

## Einer für alles: 8255

Auch der universelle Ein-/Ausgabebaustein 8255 ist an der Tastaturdekodierung beteiligt. Da dieser IC jedoch auch für Joystickanschluß, Tongenerator, Kassettenrecorderverwaltung, Druckerausgabe und Reset zuständig ist, rührt sich beim Ausfall dieses Bausteins fast gar nichts mehr. Wenn alle eben genannten Funktionen des CPCs streiken, dann ist die Diagnose leicht: Der 8255 muß gegen einen neuen Baustein ausgetauscht werden. Bei einem Preis von unter 10 Mark für das IC ist dies eine denkbar preiswerte Computer-Reparatur.

Kommen wir nun zu den schwierigsten Fehlfunktionen: denjenigen, die einen Totalausfall bewirken. In diesem Fall fehlt zuerst scheinbar jeglicher Ansatzpunkt. Doch bei planvollem Vorgehen läßt sich auch in diesem Fall etwas erreichen.

Die erste logische Vermutung bei einem Totalausfall des Computers ist der Verdacht auf Stromausfall. Also überprüfen Sie erst einmal, ob der Netzstecker richtig in der Steckdose steckt und das Monitorkabel zur Spannungsversorgung am Computer angeschlossen ist. Als nächstes überzeugen Sie sich davon, daß die Steckdose, an der Ihr CPC angeschlossen ist, auch Strom führt. Ein Blick in den Sicherungskasten klärt Sie auf.

Eine weitere Fehlerquelle sind Gewitter und Störungen im Elektrizitätswerk, die kurzzeitige Netzspannungsausfälle bewirken können. In diesem Fall ist Ihr CPC ganz und gar unschuldig.

Helfen die bis jetzt erwähnten Ratschläge nicht weiter, so müssen Sie den Monitor eines funktionsfähigen CPCs an Ihr Gerät anschließen. Erwacht Ihr Computer dadurch wieder zum Leben, liegt der Fehler im Netzteil Ihres Monitors. Dann geben Sie das Gerät bitte unbedingt in Reparatur, denn im Monitor sind noch lange nach Ausschalten und Ziehen des Netzsteckers Spannungen von mehreren 1000 Volt vorhanden.

Ist auch der Monitor an der Störung Ihres Computersystems unschuldig, so muß der Fehler direkt im Gerät liegen. Dazu ziehen Sie wieder den Netz-

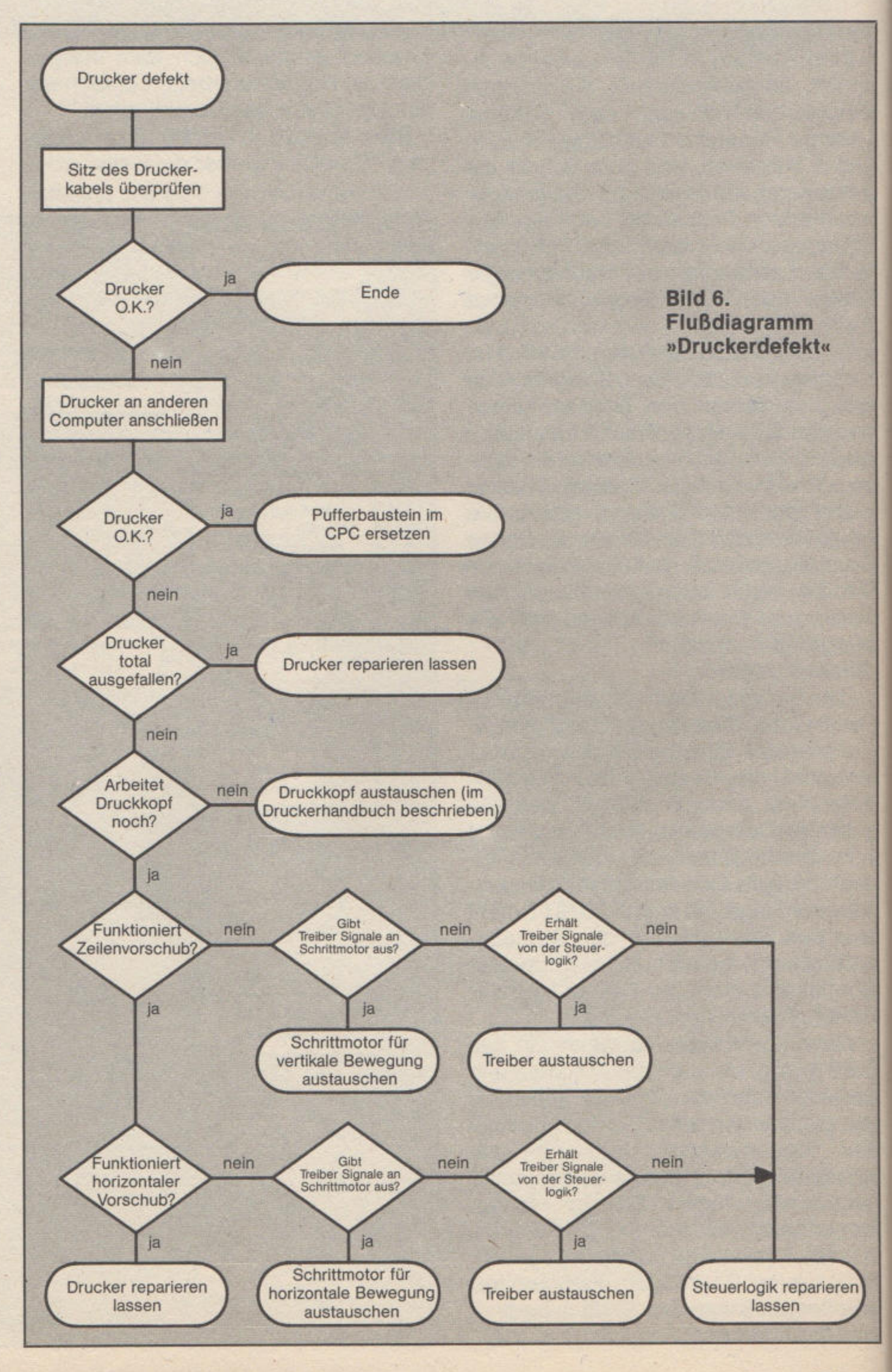



**Bild 6. Flußdiagramm »Druckerdefekt«**

stecker und zerlegen das Gerät wie oben beschrieben. Schauen Sie sich nun sehr sorgfältig die Platine Ihres CPCs an. Befindet sich irgendwo ein Löttropfen auf der Platine und schließt zwei Leitungen kurz (sehr unwahrscheinlich), oder sieht einer der kleinen braunen Kondensatoren verschmort aus, so daß er einen Kurzschluß verursachen könnte? Vielleicht finden Sie auch in einer Leiterbahn einen winzigen Haarriß, der zu einer Signalunterbrechung führt.

Diese Fehler lassen sich allesamt sehr einfach beseitigen. Der Löttropfen wird vorsichtig mit einem kleinen Messer entfernt, der Kondensator mit dem Seitenschneider abgezwickt (wenn nur ein einziger entfernt wird, muß er nicht ersetzt werden), und der Haarriß mit etwas Lötzinn geflickt.

## Des Schneiders Kern

Waren bis jetzt alle Bemühungen fruchtlos, so müssen Sie zum Kern Ihres CPCs vorstoßen. Zuerst sollten Sie den gesockelten Mikroprozessor Z80 probeweise gegen einen Ersatzbaustein austauschen. Dieses IC stellt das Herzstück des CPCs dar und ist für die Programmbearbeitung, Steuerung der Speicherverwaltung und Ein-/Ausgabe von Daten an die Peripherie zuständig. Den Baustein erhalten Sie im Handel teilweise schon für unter fünf Mark und so ist er auch für finanziell weniger gutstehende Bastler erschwinglich.

Stellt sich nach Einsetzen des Mikroprozessors heraus, daß Ihr Gerät immer noch nicht funktioniert, so ist das Gate Array eine weitere Fehlerquelle.

Das Gate Array ist ein Spezialbaustein, der eigens für die CPC-Serie konstruiert wurde und die Funktion von mehreren konventionellen Logikbausteinen in sich vereint. Das IC ist unter anderem für Speicherverwaltung, Bildausgabe, Interruptsteuerung und Erzeugung der verschiedenen Taktfrequenzen verantwortlich.

Das Überprüfen des Gate Arrays verlangt allerdings ziemlichen Aufwand. Einerseits ist der Baustein 40polig (das heißt, es müssen 40 Pins ausgelötet werden), andererseits ist er sehr teuer, so daß ein Kauf für Testzwecke nicht zu empfehlen ist. In diesem Fall empfiehlt es sich, den Fachhändler zu Rate zu ziehen. Schildern Sie ihm Ihr Problem und geben Sie den Computer in seine Hände. Es ist günstiger, einen gewissen Betrag für die Reparatur zu bezahlen, als einen Totalschaden des Gerätes zu riskieren.

Die letzte potentielle Fehlerquelle, die wir an dieser Stelle behandeln, ist ein defekter Speicherbaustein. Der Speicher des CPCs unterteilt sich in Festwertspeicher (ROM) und Arbeitsspeicher (RAM). Der Festwertspeicher enthält Betriebssystem und Basic, im Arbeitsspeicher werden Systemdaten des Betriebssystems und Programme und Daten des Anwenders abgelegt.

Der Festwertspeicher besteht aus einem einzigen Baustein, so daß das Austesten dieses ICs auf die gleiche Weise wie beim Gate Array erfolgt. Ein Defekt ist allerdings sehr unwahrscheinlich.

Bleibt als letzte Fehlerquelle der Arbeitsspeicher. Die CPCs 464 und 664 enthalten acht Speicherbausteine des Typs 4164. Wenn auch nur ein Baustein davon ausfällt, so ist, bedingt durch die interne Speicherorganisation des Computers, in sämtlichen 65536 Speicherzellen des Arbeitsspeichers ein Bit falsch gesetzt. Da das Betriebssystem wichtige Daten, wie zum Beispiel Firmware- und Restart-Vektoren im Arbeitsspeicher ablegt, führen falsche Bits an dieser Stelle zu einem sofortigen Computerabsturz.

Ein Spezialfall ist hier der CPC 6128. Er arbeitet mit 16 Speicherbausteinen. Wenn eines der unteren acht ICs seinen Geist aufgibt, führt dies gleichermaßen zu einem Computerabsturz, wie bei den anderen CPCs auch. Liegt der Defekt dagegen in einem der oberen acht Speicherbausteinen, so »hängt« sich der Computer spätestens beim Aufruf von CP/M Plus auf.

Das Austesten und Auswechseln der Speicherbausteine ist eine heikle Sache, weil diese sehr empfindlich und ungesockelt sind. Deshalb ist es ratsam, den Computer in Reparatur zu geben. Sollte zudem die Fehlerquelle wider Erwarten nicht im Arbeitsspeicher liegen, so handelt es sich um eine nur mit viel Aufwand zu lokalisierende Fehlfunktion. Lediglich der Fachmann mit der entsprechenden elektronischen Ausrüstung ist in der Lage, diese Störung zu finden und zu beseitigen.

(ma)

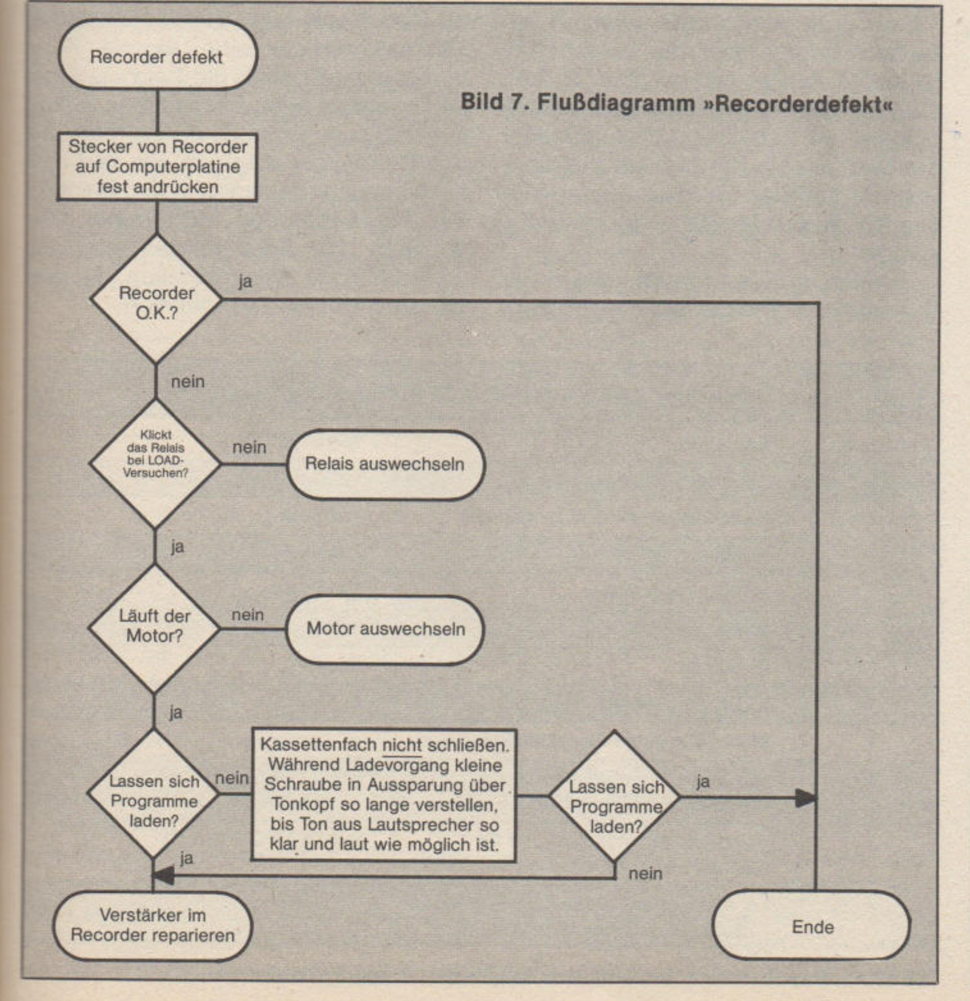



Bild 7. Flußdiagramm »Recorderdefekt«

**Ein Aufruf an alle Bastler:**

Dieser Beitrag vermittelt allgemeine Ratschläge zur Reparatur der Schneider CPC. Doch vielleicht haben auch Sie mit der Zeit den einen oder anderen Trick herausgefunden, mit dem sich bestimmte Fehler aufspüren und beheben lassen. Wir würden uns freuen, wenn Sie uns diese Tips mitteilen. Die Hinweise müssen nicht nur das Grundgerät selbst, sondern können auch die Peripherie betreffen. Alle interessanten Einsendungen werden wir in loser Folge veröffentlichen, so daß Ihr Wissen auch anderen CPC-Besitzern zugute kommt.

# Klein, aber fein

**Eine »fremde« Schaltung an Ihrem Schneider? Dazu benötigen Sie ein Interface, das die elektrischen Signale aneinander anpaßt. Eine ganz einfache Lösung, die die Dateneingabe von 8 Bit parallel erlaubt, stellt unser Minimal-Interface dar.**

Oft steht der Hobby-Bastler vor dem Problem, daß er in einem Elektronikgeschäft oder einer Zeitschrift eine interessante Schaltung entdeckt (zum Beispiel die Meßwerterfassung in dieser Ausgabe), aber ein Interface zum Anschluß der Schaltung an seinen Computer fehlt. Ein solches Interface muß erst konstruiert und aufgebaut werden; vor allem für Anfänger eine fast unüberwindliche Schwierigkeit.

Allen Lesern, die ein Interface zur parallelen Eingabe von 8-Bit-Daten benötigen, bietet das Minimal-Interface eine preiswerte und einfache Lösung. Zum Aufbau brauchen Sie lediglich drei ICs und zwei Kondensatoren. Der Gesamtpreis der Bauteile liegt damit unter 5 Mark. Rechnet man noch die Kosten für Sockel, Platine, Flachbandkabel und Computerstecker hinzu, fallen etwa 30 Mark an.

Bild 4 zeigt den Schaltplan des Minimal-Interface. Tabelle 1 listet die Bauteile und das mechanische Zubehör auf. Die Funktion des Interface verstehen Sie am besten, wenn Sie dessen Aufgabe näher untersuchen.

Das Interface muß die acht Bitsignale, die die angeschlossene Schaltung liefert, übernehmen, für die Elektronik des CPC aufbereiten und an den Computer übergeben. Hierzu bietet sich der 8-Bit-Leitungstreiber 74LS244 an, der aus acht einzelnen Treibern (Digitalverstärkern) besteht. Dieser Baustein besitzt zusätzlich die Fähigkeit, daß er sich abschalten, das heißt in den hochohmigen Zustand (Tri-State) versetzen läßt.

## Verwirrter Computer

Die Signalverarbeitung des Interface löst der 8-Bit-Treiber. Ohne äußere Beschaltung würde das IC die eingehenden Daten ständig an den Datenbus des CPC weiterreichen. Weil der Computer den Datenbus jedoch auch bei anderen Vorgängen benötigt (zum Beispiel Speicherzugriffe), würden die Daten des Interface die Elektronik des Computers gänzlich verwirren, ihn unter Umständen sogar zerstören. Aus diesem Grund ist es erforderlich und auch verständlich, daß sich der Treiber-Baustein ausschalten läßt.

Ein- und Ausschalten des 8-Bit-Treibers erfolgt über die beiden $\overline{CE}$-Eingänge (chip enable). Der Balken über der Abkürzung bedeutet, daß die Signale mit negativer Logik arbeiten, das heißt, sie sind im Zustand Null aktiv. Deshalb schaltet eine Null an den beiden Eingängen den Baustein ein, und 1 schaltet aus.

Zum Ein- und Ausschalten des 8-Bit-Treibers wird eine Decodierlogik benötigt, die den Baustein nur dann aktiviert, wenn der CPC es wünscht. Das ist einfach gesagt, aber wie kann die Decodierlogik eine Eingabe-Anforderung des Computers an das Interface erkennen?

Der Mikroprozessor Z80 des CPC erzeugt ein spezielles Signal, wenn kein Speicher-, sondern ein Portzugriff erfolgt. Ein Port stellt für den Z80 eine Art Kanal zum Austausch von Daten mit der Peripherie dar. Zur Abfrage von Tastatur und Joystick sowie zur Datenausgabe an Bildschirm und Drucker wird dieses Verfahren im CPC benutzt. Was liegt näher, als diese Methode auch für das Ansprechen des Interface zu verwenden.

Durch das Signal $\overline{IORQ}$ zeigt der Z80 an, daß ein Port angesprochen ist. Über das $\overline{RD}$-Signal wird zusätzlich ein Lesezugriff gekennzeichnet. Diese zwei Signale reichen aus, um eine Eingabe-Anforderung des Computers an das Interface anzuzeigen.

Um die einzelnen Ports, die bereits in der Hardware des CPC installiert sind, nicht miteinander zu verwechseln, spricht die Z80 jeden Port mit einer 16-Bit-Adresse an. Damit auch das Minimal-Interface eine eigene Portadresse erhält, muß die Decodierlogik des Interface neben den beiden Signalen $\overline{IORQ}$ und $\overline{RD}$ auch die Portadresse des Computers abfragen. Sie darf erst bei einer ganz bestimmten Kombination der Adreß-Bits den 8-Bit-Treiber freischalten. Die Adresse des Interface wird durch die Bitkombination, die den Baustein freischaltet, definiert.

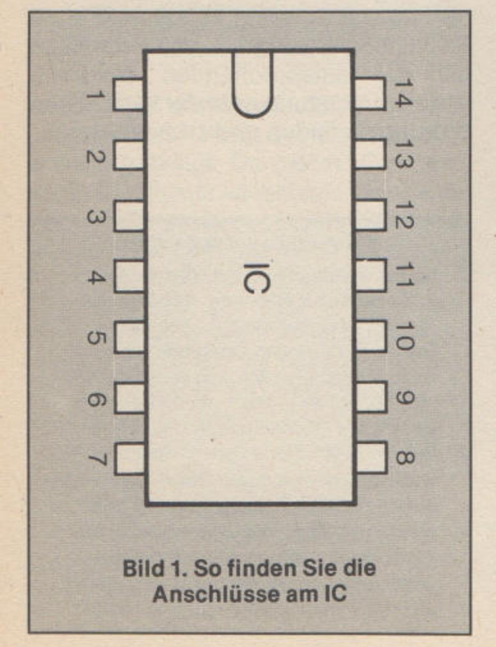



Bild 1. So finden Sie die Anschlüsse am IC

**Bauteile und Zubehör zum Minimal-Interface**

| Anzahl | Bauteil(e) | Wert/Typ |
|---|---|---|
| 2 | Keramikkondensatoren | 100 nF |
| 1 | NAND-Gatter mit acht Eingängen | 74LS30 |
| 1 | vierfaches OR-Gatter | 74LS32 |
| 1 | 8-Bit-Leitungstreiber | 74LS244 |
| 2 | IC-Sockel | 14polig |
| 1 | IC-Sockel | 20polig |
| 1 | IC-Experimentierplatine | |
| 1 | Meter 50poliges Flachbandkabel | |
| 1 | 50poliger Direktstecker für Flachbandkabel-Anschluß | nur für CPC 464/664 |
| 1 | 50poliger Amphenolstecker für Flachbandkbabel-Anschluß | nur für CPC 6128 |
| 8 | Löt-Ösen oder Steckverbindung für den Anschluß der externen Schaltung | |

Tabelle 1. Das Interface besteht aus bewährten Standardbauteilen

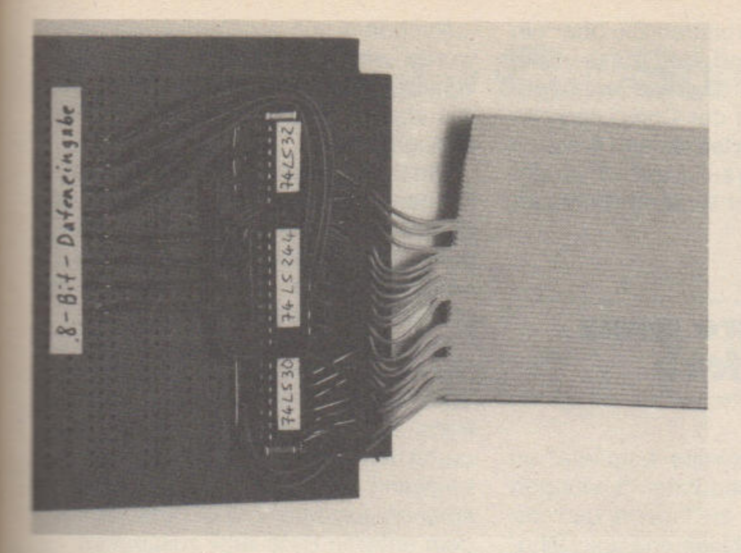



Bild 2. Von oben sieht die Schaltung etwas verwirrend aus...

Bild 3. ...doch unten sind nur Lötstellen und eine Drahtbrücke

**Die freien Port-Adressen des CPC**

| Hexadezimale Werte | Binäre Werte | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| F8E0 bis F8FF | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | X | X | X | X | X |
| F9E0 bis F9FF | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | X | X | X | X | X |
| FAE0 bis FAFF | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | X | X | X | X | X |
| FBE0 bis FBFF | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | X | X | X | X | X |
| alle aufgeführten | 1 | 1 | 1 | 1 | 1 | 0 | X | X | 1 | 1 | 1 | X | X | X | X | X |

Tabelle 2. Vier Adreßbereiche hält der CPC für den Anwender frei

Leider können Sie für das Interface keine beliebige Adresse wählen, denn ein Großteil ist bereits durch die Hardware des CPC belegt. Für den Anwender kommen lediglich die Adressen F8E0 bis F8FF hex, F9E0 bis F9FF hex, FAE0 bis FAFF hex und FBE0 bis FBFF hex in Frage.

## Aufgepaßt beim Adressieren

Tabelle 2 zeigt diese vier Adreßbereiche in Binärwerte umgewandelt. Für jeden Bereich sind die veränderlichen Bits durch ein Kreuz markiert. Die unterste Zeile der Tabelle zeigt an, welche Bits insgesamt veränderlich sind. Das Ergebnis ist, daß die Adreß-Bits A15, A14, A13, A12, A11, A7, A6 und A5 auf 1 liegen müssen, und der Wert des Adreß-Bit A10 0 betragen muß, damit nicht ein Port angesprochen wird, das vom CPC bereits reserviert ist.

Ein Blick auf den Schaltplan zeigt, daß alle genannten Adreß-Bits zur Decodierung der Interface-Adresse herangezogen werden. Alle Adreß-Bits, die auf 1 liegen müssen, sind an ein NAND-Gatter mit acht Eingängen geschaltet. Der Ausgang des NAND-Gatters geht nur dann auf Null, wenn alle Eingänge auf 1 liegen. Die Signale A10, $\overline{IORQ}$ und $\overline{RD}$, die für einen Interface-Zugriff auf Null liegen müssen, sind an OR-Gatter geschaltet. Die Ausgänge der OR-Gatter gehen im Gegensatz zum NAND-Gatter nur dann auf Null, wenn alle Eingänge auf Null liegen. Da der 8-Bit-Treiber durch ein Signal mit dem Wert Null aktiviert wird, bewirken nur die oben genannten Bit-Zustände eine Freigabe des Bausteins.

Sicherlich ist Ihnen aufgefallen, daß auch das Adreß-Bit A4 zur Decodierung der Portadresse herangezogen wird. Wenn das Interface zur Decodierung nur die Adreß-Bits verwenden würden, deren Zustand für die Unterscheidung zwischen bereits belegten und freien Ports des CPC wichtig ist, könnte das Interface durch jede der in Tabelle 2 aufgeführten Adressen angesprochen werden.

Weil der Baustein 74LS32 vier OR-Gatter enthält und somit noch ein Gatter zur Verfügung steht, diente kurzerhand ein weiteres Adreß-Bit dazu, den Adreßbereich ein wenig einzuschränken. Durch die Bedingung, daß das Adreß-Bit A4 zur Freigabe des Interface auf Null liegen muß, kann das untere Adreß-Byte nicht jeden beliebigen Wert annehmen, sondern muß im Bereich E0 bis EF hex liegen, um zusammen mit dem oberen Byte F8, F9, FA oder FB hex eine korrekte Portadresse zu bilden.

Die beiden Keramikkondensatoren zu je 100 Nanofarad in der Interface-Schaltung dienen zur Dämpfung von Schaltspannungsspitzen.

Der mechanische Aufbau des Interface erfolgt auf einer IC-Experimentierplatine. Diese Platinenart hat eine besondere Anordnung von Lötstreifen, die den Aufbau von IC-Schaltungen vereinfacht. Einige »Trockenversuche« beim Plazieren der ICs auf der Platine helfen Ihnen, die günstigsten Positionen herauszufinden, um mit einer minimalen Anzahl von Drahtbrücken auszukommen. Bild 2 und 3 zeigen Ober- und Unterseite eines Aufbaubeispiels.

## Verlötet und verkabelt

Anstelle der ICs löten Sie zwei 14polige Sockel (für 74LS30 und 74LS32) und einen 20poligen Sockel (für 74LS244) ein. So vermeiden Sie Beschädigungen der Bausteine beim Einbau. Erst dann, wenn das Interface bereits komplett aufgebaut und ohne ICs im »Leerlauf« am Computer auf Kurzschlüsse getestet wurde, stecken Sie die Bausteine in die dafür vorgesehenen Sockel.

Die Anschlüsse der ICs sind entgegen dem Uhrzeigersinn durchnumeriert. Wenn Sie den Baustein mit der längeren Seite so vor sich legen, daß sich die Einkerbung auf der linken Seite befindet, ist der Anschluß mit der Nummer 1 links auf der Ihnen zugewandten Seite (siehe Bild 1).



Die beiden Kondensatoren werden nahe den ICs zwischen +5 Volt und Masse eingelötet. Für den Anschluß der Datensignale, die das Interface verarbeiten soll, verwenden Sie entweder acht Löt-Ösen, oder eine 9polige Steckverbindung (ein Pol für Masse).

Das Interface hat einen Stromverbrauch von nur 70 Milliampere. Aus diesem Grund ist die Versorgung über eine externe Spannungsquelle nicht zwingend erforderlich, und Sie können die +5 Volt an Pin 27 des Erweiterungsanschlusses abgreifen.

In der Regel arbeitet jedoch die Schaltung, die an das Interface angeschlossen ist und die Daten liefert, mit einer eigenen 5-Volt-Spannungsquelle. Den Masseanschluß dieser Spannungsquelle müssen Sie ohnehin zum Potentialausgleich mit der Masse des Interface verbinden. Wenn Sie zusätzlich das +5-Volt-Signal an das Interface anschließen, erfolgt die Spannungsversorgung über diese Leitung.

Das Interface verbinden Sie über ein Flachbandkabel und den passenden Stecker mit dem CPC. Das Flachbandkabel legen Sie auf der Computerseite in das Steckerunterteil ein und pressen das Oberteil kräftig auf. Dadurch erfolgt ein stabiler, lötfreier Kontakt (Schneid-Klemm-Technik).

## Software ganz einfach

Auf der Interfaceseite empfiehlt es sich, die 29 unbenutzten Leitungen einige Zentimeter zu kürzen und die übrigen Signale mit kleinen Papierfähnchen zu kennzeichnen. Die Fähnchen beschriften Sie mit der Bezeichnung der Anschlüsse. Auf diese Weise ist nahezu ausgeschlossen, daß Sie die Leitungen beim Anlöten an das Interface miteinander verwechseln.

Nachdem nun ausführlich dargelegt wurde, wie das Minimal-Interface funktioniert, fehlen noch die Informationen, wie das Interface über die Software aktiviert wird.

Um Daten von einem Port einlesen zu können, stellt das Basic des CPC den Befehl »wert=INP(adresse)« zur Verfügung. In Maschinensprache dienen die beiden Befehle »LD B, ob« und »IN r,(ub)« zum gleichen Zweck.

In beiden Fällen aktiviert der Z80 die Signale $\overline{IORQ}$ sowie $\overline{RD}$ und gibt eine 16-Bit-Adresse zur Adressierung des Ports aus. Die Adresse wird in Basic durch den Wert der Variablen »adresse« bestimmt und setzt sich in Maschinensprache aus dem oberen Byte »ob« und dem unteren Byte »ub« zusammen.

In Basic wird das eingelesene Datenbyte der Variablen »wert» zugewiesen, und in Maschinensprache in das Register r geschrieben. Für die Portadresse verwenden Sie einen der oben genannten Werte. (ma)

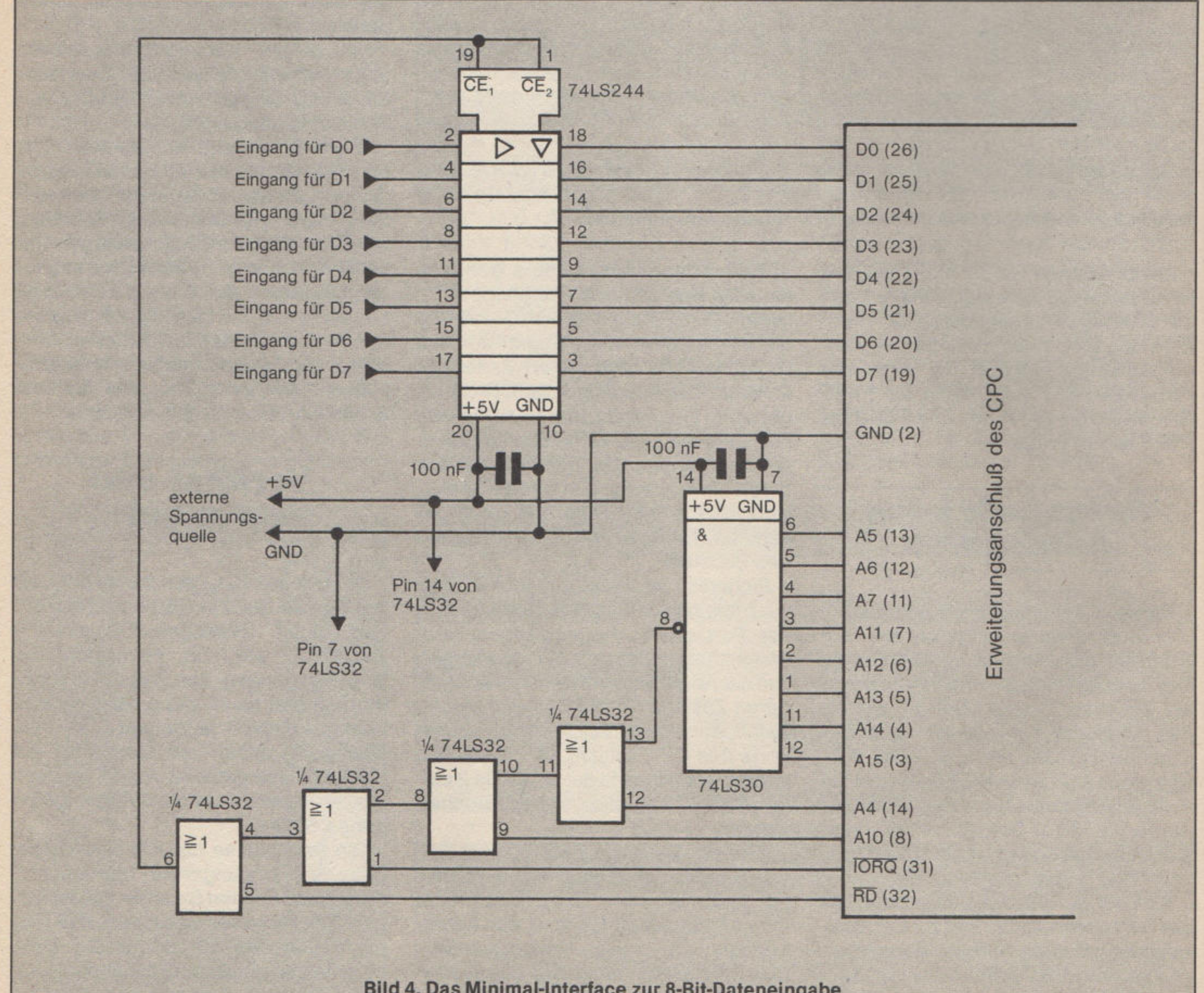
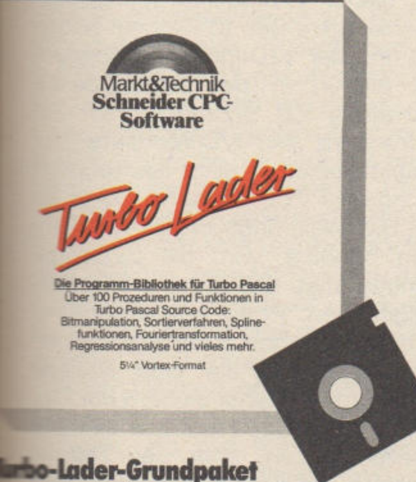



**Bild 4. Das Minimal-Interface zur 8-Bit-Dateneingabe**

Speziell für Schneider-Computer

# TURBO-LADER

## Die Programm-Bibliothek für Turbo-Pascal.










### Turbo-Lader-Grundpaket

Das Turbo-Lader-Grundmodul ist eine umfangreiche Programm-Bibliothek für den Turbo-Pascal-Programmierer. Sie umfaßt zahlreiche ausführlich dokumentierte Prozeduren und Funktionen, die der Profi zur schnellen Lösung seiner Programmieraufgaben verwenden kann und die dem Einsteiger das Erlernen der Pascal-Programmierung erleichtern.

- Bitmanipulation
- Optimale Sortierverfahren
- Anwendung von Spline-Funktionen
- Regressionsanalyse

Alle Routinen werden im kommentierten Quellcode für den Turbo-Pascal-Compiler ausgeliefert.

Software-Anforderung:
Turbo-Pascal-Compiler.

### Turbo-Lader Business

Turbo-Lader Business umfaßt einen komfortablen Bildschirm-Maskengenerator und eine professionelle Dateiverwaltung. Der Maskengenerator gibt dem Pascal-Programmierer ein Werkzeug zur einfachen Bearbeitung von Bildschirm-Masken in die Hand.

Mit diesen beiden Modulen stehen dem Anwendungsprogrammierer zwei professionelle Werkzeuge zur zeit- und kostensparenden Erstellung kommerzieller Anwendungen zur Verfügung. Alle Routinen werden im kommentierten Quellcode für den Turbo-Pascal-Compiler ausgeliefert.

Software-Anforderung:
Turbo-Pascal-Compiler, Turbo-Lader-Grundpaket

### Turbo-Lader Science

Turbo-Lader Science ist eine Sammlung technisch/wissenschaftlicher Funktionen und professioneller statistischer Verfahren für die Bereiche Medizin, Betriebs- und Volkswirtschaft, Technik und Naturwissenschaften.

- Arithmetische Operationen zur Verarbeitung komplexer Variablen
- Wichtige Funktionen: Potenz, Wurzel, trigonometrische und transzendente exponentielle Funktion
- Der Statistikteil: ein praktisches und direkt verwendbares Werkzeug zur computerunterstützten, effektiven Datenanalyse.

Software-Anforderung:
Turbo-Pascal-Compiler, Turbo-Lader-Grundpaket

| | Version | Format | Bestell-Nr. | DM | sFr | öS |
|---|---|---|---|---|---|---|
| Turbo-Lader-Grundpaket | CPC 464, 664, 6128 | 3"<br>5¼" | MS 413<br>MS 415 | 138,-*<br>138,-* | 125,-<br>125,- | 1380,-*<br>1380,-* |
| Turbo-Lader Business | CPC 464, 664, 6128 | 3"<br>5¼" | MS 423<br>MS 425 | 148,-*<br>148,-* | 132,-<br>132,- | 1480,-*<br>1480,-* |
| Turbo-Lader Science | CPC 464, 664, 6128 | 3"<br>5¼" | MS 433<br>MS 435 | 189,-*<br>189,-* | 169,-<br>169,- | 1890,-*<br>1890,-* |
| Turbo-Pascal 3.0 | CPC 464, 664, 6128, Joyce | 3"<br>3" | MS 514<br>MS 515 | 225,72*<br>225,72* | 198,-<br>198,- | 1990,-*<br>1990,-* |
| Turbo-Pascal 3.0 mit Grafikunterstützung | CPC 464, 664, 6128 | 3" | MS 524 | 285,-* | 249,- | 1990,-* |
| Turbo-Tutor (deutsch) | CPC 464,664, 6128, Joyce | 3"<br>3" | MS 534<br>MS 535 | 104,86*<br>104,86* | 92,-<br>92,- | 1190,-*<br>1190,-* |
| Turbo-Tutor (englisch) | CPC 464, 664, 6128, Joyce | 3"<br>3" | MS 544<br>MS 545 | 104,86*<br>104,86* | 92,-<br>92,- | 1190,-*<br>1190,-* |
| Turbo Graphix Toolbox | CPC 6128 | 3" | MS 564 | 225,72* | 198,- | 1990,-* |
| Turbo-Toolbox | CPC 464, 664, 6128, Joyce | 3"<br>3" | MS 554<br>MS 555 | 225,72*<br>225,72* | 198,-<br>198,- | 1890,-*<br>1890,-* |

* inkl. MwSt. Unverbindliche Preisempfehlung

Diese Markt&Technik-Software erhalten Sie in den Fachabteilungen der Warenhäuser, bei Ihrem Computerfachhändler, im Buchhandel oder direkt beim Verlag gegen Vorauskasse. Fragen Sie auch nach dem neuen Gesamtverzeichnis Herbst '86. Oder fordern Sie es direkt beim Verlag an.

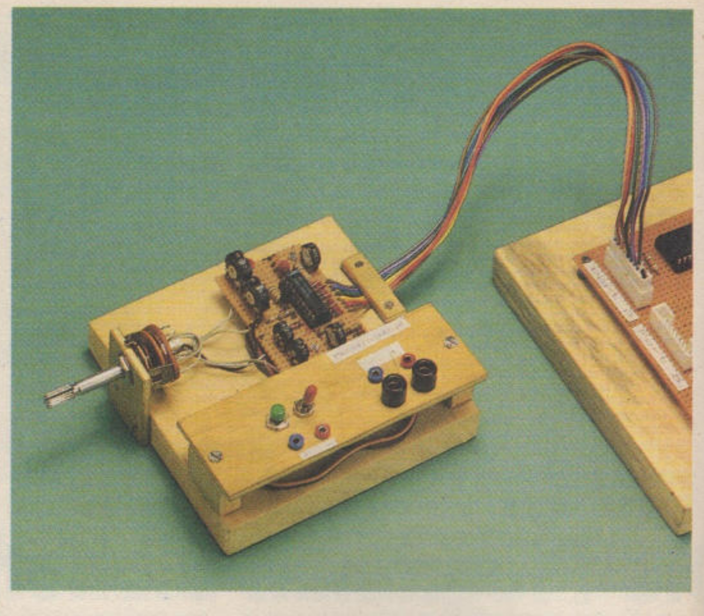



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

# Speicher-oszilloskop selbstgebaut

**Ein Speicheroszilloskop steht ganz oben auf der Wunschliste eines Hobby-Bastlers. Und ein Digitalvoltmeter oder ein Oszillograph. Wir stellen Ihnen eine preiswerte Schaltung vor, die alle drei Wünsche erfüllt.**



Eine rustikale Lösung des Schaltungsaufbaus

Das kann doch nicht wahr sein, werden Sie denken. Eine Schaltung funktioniert als Speicheroszilloskop, Digitalvoltmeter und Oszillograph zugleich und soll nur 30 Mark kosten? Es ist wahr, und mit etwas Bastelerfahrung können Sie diese Schaltung sogar nach Wunsch zu einem Digitalthermometer, einem Helligkeits- oder Feuchtigkeitsmeßgerät erweitern.

Natürlich dürfen Sie an eine Schaltung für 30 Mark keine hohen Ansprüche in bezug auf Geschwindigkeit und Genauigkeit stellen, und einen kleinen Haken hat die Sache auch, denn zum Betrieb der Schaltung ist ein einfaches 8-Bit-Interface notwendig. Dieses Interface muß Ihr CPC über eine Portadresse ansprechen und 8 Bit an dessen Datenbus übergeben. Das ist aber auch schon alles.

Das erforderliche Interface bauen Sie aus einem parallelen Interface-Baustein (8255 oder Z-80 PIO), einem 8-Bit-Leitungstreiber für den Computereingang (74LS244) und einer Decodierschaltung für die Portadresse (NAND-Gatter 74LS30, Decoder 74LS138 und einige Inverter für einzelne Adreßbits) auf.

Wenn Sie sich die Planung eines Interfaces nicht zutrauen, verwenden Sie das Happy-Interface aus Ausgabe 10/85, Seite 28, oder die Multifunktionskarte aus dem 5. Schneider-Sonderheft, Seite 26. Eine Minimallösung für Anfänger und Bastler mit schmalem Geldbeutel bietet die Einsteiger-Bastelei in dieser Ausgabe. Hier wird aus nur drei ICs ein funktionstüchtiges 8-Bit-Interface zur Dateneingabe konstruiert.

Wer den Selbstbau scheut, kann ein fertig aufgebautes Interface kaufen oder einen Bausatz erwerben und von einem befreundeten Bastler zusammensetzen lassen.

Kommen wir nun zu der eigentlichen Schaltung. Bild 1 zeigt den Schaltplan unserer Universal-Bastelei und Tabelle 1 listet die Bauteile, die Sie zum Aufbau benötigen. Das Herz der Schaltung besteht aus dem 8-Bit-A/D-Wandler ADC0804, der die eingehenden analogen Meßwerte in digitale Werte umwandelt und an den Computer ausgibt. Neben dem A/D-Wandler sind zum Betrieb nur noch acht Widerstände und Dioden, fünf Potentiometer und einige weitere Kleinteile notwendig.

Die Schaltung läßt sich in zwei Baugruppen unterteilen. Die erste Baugruppe erzeugt aus der Meßspannung die Eingangsspannung, und die zweite Baugruppe verarbeitet die Eingangsspannung zu einem digitalen Wert. Wenden wir uns zuerst der zweiten Baugruppe zu.

Der 10-Mikrofarad-Elektrolytkondensator glättet eventuelle Spannungsschwankungen. Der 10-Kiloohm-Widerstand und der 120-Pikofarad-Kondensator dienen in der angegebenen Beschaltung zur Erzeugung der Arbeitsfrequenz des A/D-Wandlers. Der Baustein arbeitet mit einer Taktfrequenz von ungefähr 760 Kilohertz. Da er für einen Digitalisierungsvorgang 64 Taktzyklen benötigt, wird die Eingangsspannung etwa 12000mal pro Sekunde digitalisiert. Profis werden kritisieren, daß sich ohne Sample-and-Hold-Glied (Schaltung, die eine analoge Spannung zwischenspeichert) nur Frequenzen bis zu 600 Hertz ausreichend darstellen lassen, aber den Hobby-Anwender wird das bei dem günstigen Preis nicht weiter stören.

Der Taster T startet auf Druck den Wandlungsvorgang, und die acht Dioden an den Ausgängen des A/D-

| Anzahl | Bauteile | Wert/Typ |
|---|---|---|
| 1 | Drucktaster | |
| 1 | Kippschalter | 1polig |
| 1 | Drehschalter | 4stufig |
| 1 | Widerstand | 220Ω |
| 2 | Widerstände | 1 kΩ |
| 1 | Widerstand | 3,3 kΩ |
| 1 | Widerstand | 6,8 kΩ |
| 2 | Widerstände | 10 kΩ |
| 1 | Metallfilm-Widerstand | 10 kΩ, 1 Prozent |
| 1 | Potentiometer | 1 kΩ |
| 4 | Potentiometer | 2,2 kΩ |
| 1 | Keramikkondensator | 150 pF, 5 Prozent |
| 1 | Elektrolytkondensator | 10 μF |
| 8 | Standarddioden | 1N4148 |
| 1 | Germaniumdiode | AA 136 oder ähnliches |
| 1 | Zenerdiode | ZPY 5,6 |
| 1 | 8-Bit-A/D-Wandler | ADC0804 oder ADC0802 |

*Tabelle 1. Das brauchen Sie. Vergleichen Sie die Preise bei A/D-Wandlern.*

Wandlers schützen den Baustein vor Spannungsspitzen. Mit dem Potentiometer an Pin 9 des A/D-Wandlers wird die Referenzspannung auf exakt 2,5 Volt justiert. Diese Einstellung begrenzt den Spannungsbereich, den der A/D-Wandler digitalisiert, auf 0 bis +5 Volt.

Einer Eingangsspannung von 0 Volt wird 0 und einer Eingangsspannung von +5 Volt der Wert 255 zugeordnet. Die Spannungswerte dazwischen erhalten den entsprechenden digitalen Gegenwert nach der Formel:

```
wert = INT(51 * spannung)
```

Da der A/D-Wandler den Spannungsbereich von 5 Volt in 255 Schritte auflöst, ergibt sich eine Auflösung von 20 Millivolt für den Meßvorgang.

Höhere Spannungen als +5 Volt verarbeitet der A/D-Wandler zwar auch, aber ein gößerer Wert als 255 läßt sich mit 8 Bit nicht darstellen. Deshalb ist eine Eingangsspannung von über +5 Volt nicht sinnvoll. Spannungen über +5,6 Volt schließt ohnehin die Zenerdiode ZPY 5,6 nach Masse kurz.

Die erste Baugruppe erzeugt aus der Meßspannung an U+ über Spannungsteiler die Eingangsspannung für den A/D-Wandler. Die anliegende Meßspannung wird über eine Leitungsbrücke mit dem gewünschten Meßbereich ($U_5$ bis 5 Volt, $U_{10}$ bis 10 Volt etc.) verbunden. Der Meßbereich wird mit dem Drehschalter DS ausgewählt.

Jeder Meßbereich läßt sich eichen, indem Sie eine bekannte Spannung anlegen und das Potentiometer des Spannungsteilers solange drehen, bis das Ergebnis der Software (Funktion 4 für Voltmeter wählen) mit dem tatsächlichen Wert übereinstimmt. Achten Sie darauf, daß Sie beim Eichen eines Meßbereichs auch im Programm den richtigen Meßbereich einstellen (Funktion 0).

Die Zenerdiode sorgt dafür, daß bei Anschlußfehlern eine für den A/D-Wandler schädliche Überspannung nach Masse kurzgeschlossen wird.

## 40 Volt Wechselspannung

Wenn Sie mit der Schaltung Wechselspannungen messen möchten, darf die Eingangsspannung an $U_{in}(+)$ des A/D-Wandlers nicht unter die Spannung an $U_{in}(-)$ fallen, andernfalls besteht die Gefahr, daß der Baustein beschädigt wird. Zum Schutz des ICs schließt die Germaniumdiode AA 136 negative Spannungen über 0,3 Volt kurz. Um trotzdem negative Wechselspannungen zu messen, behelfen Sie sich mit einem kleinen Trick. Über den Schalter S prägen Sie eine Gleichspannung, die von der Versorgungsspannung abgegriffen wird, auf die Eingangsspannung auf. Dadurch wird der Wert der Eingangsspannung angehoben und negative Werte gelangen in den positiven Bereich.

Sie müssen allerdings sehr genau auf die Einstellung des Potentiometers P achten, damit auch die negative Spitzenspannung der Meßspannung am Eingang $U_{in}(+)$ nicht unter 0 Volt fällt. Es empfiehlt sich dringend, zu Meßbeginn das Potentiometer an den oberen Anschlag zu drehen, so daß der Widerstand kurzgeschlossen wird. Bei einem periodischen Signal läßt sich die optimale Einstellung durch Verstellen des Potentiometers bei gleichzeitiger Ausgabe der Meßkurve auf den Computer-Bildschirm erzielen. Trotzdem sollten Sie für Spannungsschwankungen immer noch einen kleinen »Reserveabstand« nach unten einhalten.

Eine Wechselspannung dürfen Sie nur an den dafür vorgesehenen Meßbereich anschließen (im Zweifelsfall an $U_{40}$), und niemals mit größeren Spannungen als 42 Volt arbeiten, weil sonst die Bastelei bei Berühren von leitenden Teilen lebensgefährlich wird!

Für eine rein qualitative Darstellung von Wechselspannungen sind außer dem Schließen des Schalters S und Justieren des Potentiometers P keinerlei zusätzliche Maßnahmen nötig. Wenn Sie dagegen eine quantitative Auswertung der Signale wünschen, müssen Sie die positive und negative Spitzenspannung kennen, und ausgehend von diesen Daten in der Programmroutine zum Zeichnen des Koordinatenkreuzes eine Nullinie definieren, so daß alle Spannungswerte unterhalb dieser Linie als negative Werte dargestellt werden. Eine Anpassung der Digitalvoltmeter-Funktion an Wechselspannungen ist aufgrund des ständig schwankenden Wertes nicht sinnvoll.

## Korrekte Konstruktion

Die Spannung an U− beziehungsweise $U_{in}(-)$ stellt das Bezugspotential zur Meßspannung dar. Hier schließen Sie die Masse der Meßspannung, beziehungsweise die Spannung, die vom A/D-Wandler als Masse definiert werden soll, an. Die Stromversorgung der Schaltung darf nicht über den Computer, sondern muß über eine externe 5-Volt-Spannungsquelle erfolgen. Zum Potentialabgleich ist es jedoch erforderlich, daß Sie den Minuspol der Spannungsquelle mit dem Masseanschluß des Computers (GND) verbinden.

Für die Anhänger von großen Genauigkeiten kommt neben dem A/D-Wandler-Baustein ADC0804 auch der doppelt so genaue und etwas teurere ADC0802 in Frage. Doppelt so genau heißt in diesem Fall jedoch nicht, daß der ADC0802 mit einer Auflösung von 10 Millivolt arbeitet, sondern daß die größten Meßabweichungen durch Produktionstoleranzen nicht ein Bit (in unserem Fall 20 Millivolt), sondern nur ein halbes Bit (10 Millivolt) betragen.

Der Aufbau der Schaltung erfolgt zweckmäßigerweise auf einer IC-Experimentierplatine oder auf einer Experimentierplatine mit Lötstreifenraster. Den A/D-Wandler löten Sie nicht direkt ein, sondern bauen einen 20poligen IC-Sockel in die Schaltung. Setzen Sie den Baustein erst ein, nachdem Sie die

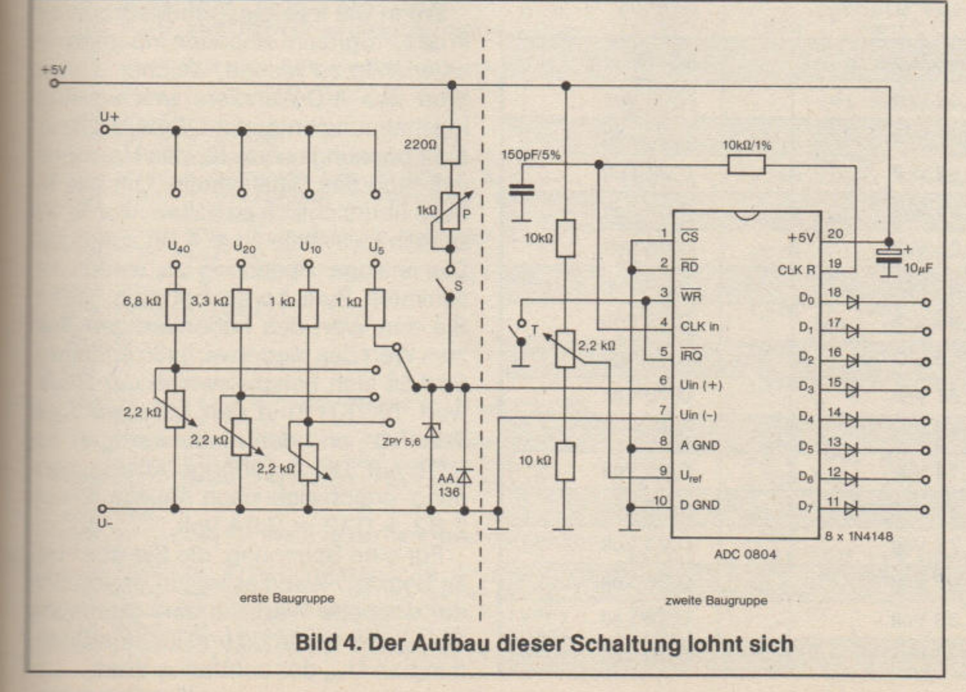
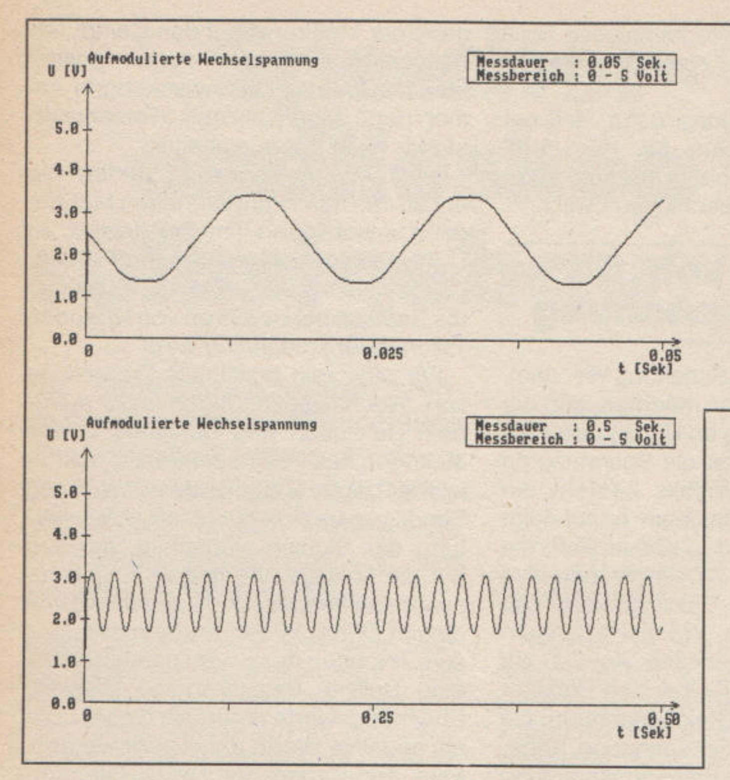



**Bild 4. Der Aufbau dieser Schaltung lohnt sich**



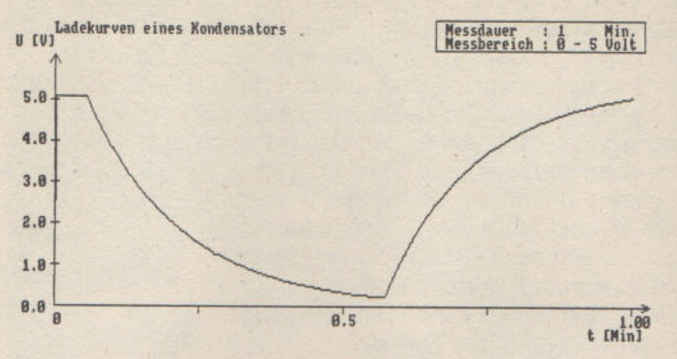




**Bild 2. Drei Meßprotokolle zur Demonstration**

Schaltung ohne A/D-Wandler an die Versorgungsspannung angeschlossen und auf Kurzschluß überprüft haben.

Für die Verbindung der Schaltung mit dem Interface sollten Sie eine solide Steckverbindung wählen, um einen sicheren Kontakt und die Mobilität im Reparaturfall zu gewährleisten.

Wenn Sie die Schaltung mit dem Interface in Betrieb nehmen, müssen Sie zuerst immer die Schaltung und dann den Computer einschalten. Beim Ausschalten geht es dann genau andersherum.

Bis jetzt können Sie mit Ihrer Schaltung noch nicht viel anfangen, weil die Software fehlt. Listing 1 zeigt Ihnen das Programm, das die Schaltung in ein Speicheroszilloskop, Digitalvoltmeter und einen Oszillographen verwandelt. Bei Listing 2 handelt es sich um den zugehörigen DATA-Lader für drei RSX-Befehle, die Hardcopy-Routine und die Meßwerterfassung in Maschinensprache. Wenn Sie beim Eintippen der Listings im Text unterstrichene Zahlen und Buchstaben antreffen, müssen Sie die in der Anleitung zu »Explora« erwähnten Hinweise berücksichtigen.

Als Portadresse wurde im DATA-Lader F8E0 (Adresse vom Happy-Interface und Minimal-Interface) eingesetzt. Unter den Adressen 9EEB, 9F36 und 9F52 können Sie die Portadresse im DATA-Lader gegebenenfalls gegen die Adresse austauschen, die Ihr Interface verwendet. Beachten Sie bitte, daß Sie in der ersten Speicherzelle das untere Byte und an zweiter Stelle das obere Byte der Adresse ablegen müssen.

Das Hauptprogramm ist menügesteuert. Die Funktionen 0, 1 und 6 bis 9 dienen zur Voreinstellung von Standardparametern. In der Funktion 2 und 3 arbeitet das Programm als Speicheroszilloskop, indem es die Meßspannung als Meßpunkte beziehungsweise Meßkurve darstellt. Die Funktion 4 ruft das Digitalvoltmeter auf, und Funktion 5 repräsentiert den Oszillographen durch Ausgabe der Meßkurve auf den Drucker.

## Einige Anwendungen

Um Ihnen einen Eindruck von der Funktion der Schaltung und Software zu vermitteln, zeigt Bild 2 einige Meßbeispiele, die mit der Hardcopy-Routine auf einem Star-Drucker ausgedruckt wurden.

Wenn Sie Ihre eigenen Meßroutinen in das Programm einbauen möchten, ist es wichtig zu wissen, welcher digitale Wert des A/D-Wandlers welcher Analogspannung entspricht. Tabelle 2 zeigt die Spannungswerte für das Hexadezimal- und das Binärsystem. Um die Tabelle übersichtlich zu halten, wurde ein Byte in zwei Teile zu je 4 Bit aufgeteilt. Die analoge Spannung zu einem bestimmten Byte berechnet sich, indem Sie den Wert des höherwertigen Teils zum Wert des niederwertigen Teils addieren. So teilt sich beispielsweise der Binärwert 10010110 in den höherwertigen Teil 1001 und den niederwertigen Teil 0110 auf. Die zugehörige Analogspannung ergibt sich nach Tabelle 2 aus $2{,}82 + 0{,}12 = 2{,}94$ Volt.

Für eine Spannung, die Sie über $U_{10}$ an den A/D-Wandler legen, ergibt sich der doppelte Wert, für eine Spannung an $U_{20}$ der vierfache und für eine Spannung an $U_{40}$ der achtfache Wert.

(Jörg Braun/ma)

| Hex | Binär | höherwertig | niederwertig |
|---|---|---|---|
| 0 | 0000 | 0,00 Volt | 0,00 Volt |
| 1 | 0001 | 0,31 Volt | 0,02 Volt |
| 2 | 0010 | 0,63 Volt | 0,04 Volt |
| 3 | 0011 | 0,94 Volt | 0,06 Volt |
| 4 | 0100 | 1,25 Volt | 0,08 Volt |
| 5 | 0101 | 1,57 Volt | 0,10 Volt |
| 6 | 0110 | 1,88 Volt | 0,12 Volt |
| 7 | 0111 | 2,19 Volt | 0,14 Volt |
| 8 | 1000 | 2,51 Volt | 0,16 Volt |
| 9 | 1001 | 2,82 Volt | 0,18 Volt |
| A | 1010 | 3,14 Volt | 0,20 Volt |
| B | 1011 | 3,45 Volt | 0,22 Volt |
| C | 1100 | 3,76 Volt | 0,24 Volt |
| D | 1101 | 4,08 Volt | 0,26 Volt |
| E | 1110 | 4,39 Volt | 0,28 Volt |
| F | 1111 | 4,70 Volt | 0,30 Volt |

**Tabelle 2. So rechnen sich Bits in Spannung um**

```
10 '***********************************
   ***                                    [431C]
30 '                                      [5F54]
40 'Version 1.0                           [C680]
50 '                                      [6158]
60 MEMORY &9EDF:LOAD"!ADWANDEL.BIN":CALL
   &9F05:REM RSX installieren             [5D30]
70 '                                      [835C]
80 n=1:REM Variable f. Messbereich (Grun
   deinstellung 0-5V)                     [6852]
90 DIM ZEIT(20):REM Moegliche Messzeiten
                                          [8D08]
100 DIM timee%(20):REM Dazugehoerige Kon
    stanten                               [84BA]
110 '                                     [DDB2]
120 FOR a%=1 TO 19                        [4CF6]
130 READ zeit(a%)                         [3206]
140 NEXT a%                               [D934]
150 DATA 150,90,60,30,15,10,5,3,2,1,30,2
    0,10,5,1,0.5,0.1,0.05,0.03            [6628]
160 REM Dies sind die moeglichen Messzei
    ten                                   [CBC4]
170 '                                     [03BE]
180 FOR a%=1 TO 20                        [39F2]
190 READ timee%(a%)                       [EB0C]
200 NEXT                                  [D0E2]
210 DATA 150,90,60,30,15,10,5,31500,2100
    0,10500,5250,3500,1750,900,186,91,15
    ,6,1,9,7                              [7B86]
220 REM dies sind die fuer die versch. M
    esszeiten vom Mcode benoetigten Verz
    oegerungswerte.                       [84B4]
230 '                                     [07B8]
240 '*** Grundmenue ***                   [1866]
250 '                                     [E1BC]
260 CALL &9FD9:REM Alten Bildschirmaufba
    u merken (LDDR)                       [C72C]
270 MODE 2                                [9360]
280 LOCATE 1,5:PRINT STRING$(80,"_")      [331C]
290 LOCATE 2,3                            [0548]
300 PRINT"M e s s p r o g r a m m<3>G r
    u n d m e n u e                       [28B2]
310 LOCATE 1,8:PRINT"Bitte waehlen        [E10C]
320 LOCATE 20,8:PRINT"Messbereich waehle
    n ..............[ 0 ]"                [3B30]
330 LOCATE 20,10:PRINT"Messdauer einstel
    len .............[ 1 ]"               [6546]
340 LOCATE 20,12:PRINT"Darstellung durch
    Messpunkte .....[ 2 ]"                [458A]
350 LOCATE 20,14:PRINT"Darstellung als K
    urve ............[ 3 ]"               [F226]
360 LOCATE 20,16:PRINT"Voltmeter .......
    .................[ 4 ]"               [42DE]
370 LOCATE 20,18:PRINT"Kurve auf Drucker
    ausgeben .......[ 5 ]"                [D4CC]
380 LOCATE 20,20:PRINT h$"Messtart durch
    Messwertaenderung..[6/7]"h$           [DC64]
390 LOCATE 20,22:PRINT h1$"Wechselspannu
    ngsmessbereich ......[8/9]"h1$        [C6D2]
400 LOCATE 20,23:PRINT"________________
    ______________________"               [1B06]
410 LOCATE 20,25:INPUT"Gewaehlter Menuep
    unkt ............. ";mp               [3A24]
420 mp=mp+1                               [FE12]
430 ON MP GOSUB 1650,910,1160,1190,1920,
    1580,1780,1820,1860,1890              [9C58]
440 GOTO 260:REM Grundmenue darstellen    [E39E]
450 '                                     [E5C0]
460 '*** Aufbau des Koordinatenkreuzes *
    **                                    [A4E8]
470 '                                     [E3C4]
480 MODE 2                                [5166]
490 ORIGIN 40,40                          [9D32]
500 MOVE 0,0:DRAWR 0,300:REM Y-Achse      [1012]
510 MOVE 0,0:DRAWR 600,0:REM X-Achse      [8618]
520 MOVE-5,295:DRAWR 5,10:DRAWR 5,-10:RE
    M Pfeilspitzen                        [2E78]
530 MOVE 590,-5:DRAWR 10,5:DRAWR-10,5     [E100]
540 FOR a%=1 TO 5                         [E998]
550 MOVE 145*a%,-8:REM X Einteilung       [AE6C]
560 DRAWR 0,8                             [45DE]
570 MOVE-4,50*a%:                         [359E]
580 DRAWR 8,0                             [67E2]
590 NEXT                                  [71FA]
600 LOCATE 1,3:PRINT"U [V]                [3E14]
610 LOCATE 73,25                          [BE18]
620 IF wert%>10 THEN PRINT"t [Sek]"ELSE
    PRINT"t [Min]                         [2508]
630 LOCATE 75,24:PRINT USING"###.##";zei
    t(wert%):REM x Achse beschriften      [2D66]
640 LOCATE 40,24:PRINT zeit(wert%)/2      [5588]
650 LOCATE 5,24:PRINT 0                   [0DE2]
660 TAG                                   [1230]
670 IF uflag=1 THEN 810:REM Wechselspann
    ung                                   [B9C8]
680 MOVE-40,4:PRINT USING"##.#";0;:       [F91C]
690 MOVE-40,55:PRINT USING"##.#";1*n;     [8C48]
700 MOVE-40,106:PRINT USING"##.#";2*n;    [7894]
710 MOVE-40,157:PRINT USING"##.#";3*n;    [02A4]
720 MOVE-40,208:PRINT USING"##.#";4*n;    [71A2]
730 MOVE-40,259:PRINT USING"##.#";5*n;    [41B2]
810 TAGOFF                                [89E0]
820 MOVE 382,347:DRAWR 0,-36:DRAWR 210,0
    :REM Naehere Messangaben              [2D0E]
830 DRAWR 0,36:DRAWR-210,0                [9AEC]
840 LOCATE 55,2:PRINT"Messdauer<3>:"Zeit
    (wert%);"KH"                          [AC20]
850 LOCATE 75,2:IF wert%>10 THEN PRINT"S
    ek."ELSE PRINT"Min."                  [5196]
860 IF uflag=0 THEN LOCATE 55,3:PRINT"Me
    ssbereich : 0 -"5*n"Volt":GOTO 880    [F79C]
870 LOCATE 55,3:PRINT"Messbereich :"(-2.
    5)*n"-"(2.5)*n"V"                     [FBC6]
880 MOVE 0,0                              [C346]
890 RETURN                                [AF42]
900 '                                     [0EC0]
910 '*** benoetigte Eingaben ***          [ECF8]
920 '                                     [10C4]
930 MODE 2                                [CC66]
940 PRINT"Zum Anwaehlen der gewuenschten
    Messdauer"                            [9BA2]
950 PRINT"die Tasten [ "CHR$(240)" ] bzw
    . [ "CHR$(241)" ] druecken.           [4BEE]
960 MOVE 500,0:DRAWR 0,400:REM Bildaufte
    ilung                                 [0F7C]
970 MOVE 0,365:DRAWR 640,0                [A0C2]
980 LOCATE 68,2:PRINT"Messdauer "         [0A72]
990 FOR wert%=1 TO 19:REM moegliche Zeit
    en ausgeben                           [D19E]
1000 LOCATE 64,wert%+4                    [372A]
1010 PRINT USING"###.##";Zeit(wert%);     [A400]
1020 IF wert%<=10 THEN PRINT" Min."ELSE
     PRINT" Sek."                         [B120]
1030 NEXT                                 [F346]
1040 LOCATE 1,23:PRINT"Eingabe mit ENTER
     abschliessen.                        [38BA]
1050 wert%=10                             [6ED6]
1060 WHILE INKEY(18)=-1                   [C1AA]
1070 IF INKEY(0)<>-1 AND wert%>=2 THEN w
     ert%=wert%-1:REM Zeiger erhoehen     [C502]
1080 IF INKEY(2)<>-1 AND wert%<=18 THEN
     wert%=wert%+1:REM Zeiger erniedrige
     n                                    [7FEA]
1090 LOCATE 56,wert%+4:PRINT CHR$(154)CH
     R$(243)                              [E2BA]
1100 LOCATE 56,wert%+3:PRINT"<2>":REM Al
     ten Pfeil loeschen                   [9828]
1110 LOCATE 56,wert%+5:PRINT"<2>"         [5AC8]
1120 CALL &BD19:CALL &BD19:REM etwas ver
     zoegern                              [9016]
1130 WEND                                 [2B26]
1140 CALL &BB03:REM Tastenpuffer leeren   [60FE]
1150 RETURN                               [848E]
1160 POKE &9F7B,&EA:POKE &9FA3,&EA:REM Z
     eiger auf Plotroutine                [9BD8]
1170 GOSUB 1220                           [269C]
1180 RETURN                               [DA94]
1190 POKE &9F7B,&F6:POKE &9FA3,&F6:REM Z
     eiger auf Drawroutine                [2C94]
1200 GOSUB 1220                           [5590]
1210 RETURN                               [AD88]
1220 '                                    [B718]
1230 '*** Messen und Werte einlesen ***   [866A]
1240 '                                    [911C]
1250 POKE &9FFC,0:POKE &9FFD,0:REM X-Koo
     rdinate auf Null setzen              [01F0]
1260 IF wert%>7 AND wert%<15 THEN 1530:R
     EM Zeitspanne ist kleiner als 1 Min
     ute (wert% als Pointer )             [987A]
1270 IF wert%>14 THEN 1480:REM Messzeit
     <=1 Sekunde -> Maschinensprache Aus
     wertung                              [4D4C]
1280 GOSUB 450:REM Koordinatensystem auf
     bauen                                [40D8]
1290 EVERY timee%(wert%)*3000/580,3 GOSU
     B 1380:REM Nur Wert einlesen         [7DA8]
1300 CALL &BB03:REM Tastenpuffer entleer
     en                                   [6B88]
1310 IF x%<=580 THEN 1320 ELSE PRINT"G":
     CALL &BB06:REM Aufenthaltsschleife
     ELSE: Auf Anwender warten            [220E]
1320 IF INKEY(66)=0 THEN PRINT"G":CALL &
     BB06:GOTO 1340:REM ESC gedrueckt ?   [FE7A]
1330 GOTO 1310:REM in die Aufenthaltssch
     leife                                [B29C]
1340 PRINT REMAIN(3):REM Timer abschalte
     n                                    [1592]
1350 RETURN                               [AA92]
1360 '                                    [B522]
1370 '*** Messwert einlesen ***           [8236]
1380 CALL &9F91:x%=x%+1:REM Nur einen We
     rt einlesen und gleich ausgeben      [29F0]
1390 RETURN                               [8A9A]
1400 '                                    [B918]
1410 REM *** Messen durch BASIC-Timing u
     nd Maschinencodeplot ***             [DFAA]
1420 '                                    [971C]
```

**Listing 1. Das Hauptprogramm zur Meßwerterfassung**

```
1430 GOSUB 460:REM Koordinatenkreuz aufb
     auen                                   [81EC]
1440 CALL &BB03:REM ...                     [5FF4]
1450 IF bedflag=0 THEN|MESS,timee%(wert%
     )ELSE CALL &9EE6,1,timee%(wert%)REM
      Test auf Messstartbedingung-Werte
     einlesen                               [D592]
1460 PRINT"G":CALL &BB06:REM Auf Anwende
     r warten                               [5F50]
1470 RETURN                                 [8F98]
1480 REM  *** Messzeit <= 1 Sekunde ***     [5EF4]
1490 GOSUB 460:REM Koordinatenkreuz aufb
     auen                                   [5FF8]
1500 IF bedflag=1 THEN|BMESS,127,timee%(
     wert%)ELSE CALL &9F41,timee%(wert%)
     :REM Bedingung  f. Messtart ? --Wer
     te einlesen                            [540E]
1510 PRINT"G":CALL &BB06:REM Auf Benutze
     r warten                               [757E]
1520 RETURN                                 [CA90]
1530 GOSUB 460:REM Koordinatenkreuz aufb
     auen                                   [7FEE]
1540 CALL &BB03:REM Tastenpuffer leeren     [9506]
1550 IF bedflag=0 THEN|DMESS,timee%(wert
     %)ELSE CALL &9EE6,2,timee%(wert%)      [419E]
1560 PRINT"G":CALL &BB06:REM Auf Anwende
     r warten                               [CF52]
1570 RETURN                                 [9C9A]
1580 MODE 2                                 [68CA]
1590 CLS:LOCATE 10,10:INPUT"GBitte Titel
      eingeben ";titel$                     [6CFC]
1600 IF LEN(titel$)>30 THEN PRINT"Nicht
     mehr als 30 Zeichen":GOTO 1590         [DAB4]
1610 CALL &9FE5:REM Screen holen            [B82E]
1620 LOCATE 6,2:PRINT titel$                [D904]
1630 CALL &A000:REM Hardcopy                [212A]
1640 RETURN                                 [FB96]
1650 REM Messbereich aendern                [AAAE]
1660 MODE 2                                 [93C8]
1670 LOCATE 1,2:PRINT"M E S S B E R E I
     C H<3>E I N S T E L L E N"             [5C7A]
1680 MOVE 0,350:DRAWR 640,0                 [4114]
1690 LOCATE 30,8:PRINT"Messbereich 1: [
     0 -<2>5V ]"                            [77DE]
1700 LOCATE 30,10:PRINT"Messbereich 2: [
      0 - 10V ]"                            [743A]
1710 LOCATE 30,12:PRINT"Messbereich 3: [
      0 - 20V ]"                            [EF44]
1720 LOCATE 30,14:PRINT"Messbereich 4: [
      0 - 40V ]"                            [E150]
1730 LOCATE 30,15:PRINT"________________
     _________"                             [2ED2]
1740 LOCATE 1,17:INPUT"Bitte Nummer des
     gewuenschten Messbereichs eingeben
     ",n                                    [6E86]
1750 IF n=4 THEN n=8                        [149A]
1760 IF n=3 THEN n=4                        [7092]
1770 RETURN                                 [8A9E]
1780 REM Messstartbedingung ein             [3792]
1790 bedflag=1                              [C748]
1800 h$="X":REM Inverse Darstellung         [5C9A]
1810 RETURN                                 [BF94]
1820 REM Messstartbedingung aus             [18A2]
1830 bedflag=0                              [B53C]
1840 h$=CHR$(0):REM normale Darstellung     [6112]
1850 RETURN                                 [9F9C]
1860 uflag=1:REM Messbereich Wechselspan
     nung                                   [32D2]
1870 h1$="X":REM inverse Darstellung        [2D4A]
1880 RETURN                                 [A1A2]
1890 uflag=0:REM Wechselspannungsmessber
     eich aus                               [D38E]
1900 h1$=CHR$(0):REM inverse Darstellung
      aus                                   [6E5C]
1910 RETURN                                 [7C96]
1920 REM *** Voltmeter ***                  [88DC]
1930 MODE 2                                 [4CC8]
1940 PRINT:PRINT"<2>V O L T M E T E R"      [3D10]
1950 PRINT"<2>"STRING$(18,"=")              [ABC4]
1960 ORIGIN 320,200                         [7256]
1970 MOVE-80,0:DRAWR 180,0:DRAWR 0,-30:D
     RAWR-180,0:DRAWR 0,30                  [CE46]
1980 LOCATE 1,25:PRINT"Abbruch: bel.Tast
     e                                      [72D4]
1990 LOCATE 3,5:PRINT"Messbereich:<4>Vol
     t"                                     [3E66]
2000 LOCATE 15,5:PRINT n*5:REM momentane
     r Messbereich                          [7C66]
2001 LOCATE 55,5:PRINT"Moegliche Messber
     eiche:":PRINT                          [471C]
2002 PRINT TAB(68)"0-05 Volt":PRINT TAB(
     68)"0-10 Volt                          [F028]
2003 PRINT TAB(68)"0-20 Volt":PRINT TAB(
     68)"0-40 Volt                          [892A]
2005 POKE &B1C8,0:POKE &B1CF,&F0:POKE &B
     1D0,&F:REM Umschaltung nach Mode 0     [357E]
2010 EVERY 25,1 GOSUB 2080:REM Einlesen     [979A]
2020 WHILE taste$=""                        [9886]
2030 taste$=INKEY$                          [3F16]
2040 LOCATE 9,14:PRINT USING"##.##";span
     nung                                   [BE30]
2050 WEND                                   [3A2A]
2060 taste$="":PRINT REMAIN(1):REM Timer
      abschalten                            [1A8E]
2070 RETURN                                 [9F92]
2080 spannung=INP(&F8E0)*0.02*n:REM Span
     nung einlesen                          [DEBE]
2090 RETURN                                 [FB96]
```

**Listing 1. Das Hauptprogramm zur Meßwerterfassung (Schluß)**

```
100 '***************************************
    *                                       [F97C]
101 '* ADWANDEL.DAT - DATA-Lader von 'CPC'
    *                                       [4C0E]
102 '***************************************
    *                                       [7B80]
103 '                                       [DEB6]
104 DATA 9EE0,00,00,00,00,00,01,CD,1B,0185  [72EE]
105 DATA 9EE8,BB,D8,01,E0,F8,ED,78,FE,61DA  [0574]
106 DATA 9EF0,7F,28,F7,DD,7E,02,FE,01,24B5  [4108]
107 DATA 9EF8,CA,41,9F,FE,02,CA,A8,9F,6BB7  [5578]
108 DATA 9F00,C9,00,00,00,00,01,0F,9F,6405  [1FFE]
109 DATA 9F08,21,29,9F,C3,D1,BC,C9,1A,00E0  [F8CE]
110 DATA 9F10,9F,C3,2E,9F,C3,41,9F,C3,7591  [A4D2]
111 DATA 9F18,A8,9F,42,4D,45,53,D3,4D,7DDF  [52F8]
112 DATA 9F20,45,53,D3,44,4D,45,53,D3,2B69  [EE5C]
113 DATA 9F28,00,29,9F,0F,9F,C9,F5,CD,1FAB  [5436]
114 DATA 9F30,1B,BB,DA,A6,9F,01,E0,F8,37A4  [7706]
115 DATA 9F38,DD,66,02,ED,78,BC,20,EF,780F  [E518]
116 DATA 9F40,F1,FE,00,C8,F3,DD,7E,00,4F90  [4FFC]
117 DATA 9F48,32,FE,9F,DD,7E,01,32,FF,3BDF  [D95C]
118 DATA 9F50,9F,01,E0,F8,21,10,27,11,5D57  [D856]
119 DATA 9F58,44,02,ED,78,77,23,1B,D5,3B77  [96A2]
120 DATA 9F60,CD,87,9F,D1,7A,B3,20,F2,581E  [F4E4]
121 DATA 9F68,11,00,00,01,10,27,21,44,088A  [D6C0]
122 DATA 9F70,02,E5,0A,26,00,6F,03,13,3AC9  [7A52]
123 DATA 9F78,D5,C5,CD,EA,BB,C1,D1,E1,4B5F  [A678]
124 DATA 9F80,2B,7C,B5,20,EC,FB,C9,ED,1BD3  [C954]
125 DATA 9F88,5B,FE,9F,7A,B3,C8,1B,18,00D6  [CF18]
126 DATA 9F90,FA,01,E0,F8,ED,78,ED,5B,69C9  [7E4A]
127 DATA 9F98,FC,9F,13,ED,53,FC,9F,26,5500  [EB0E]
128 DATA 9FA0,00,6F,CD,EA,BB,C9,F1,C9,0B17  [F44E]
129 DATA 9FA8,01,44,02,FE,01,C0,AF,32,1C44  [EA98]
130 DATA 9FB0,FE,9F,32,FF,9F,32,FC,9F,5427  [7648]
131 DATA 9FB8,32,FD,9F,DD,7E,00,32,FE,3B1A  [293A]
132 DATA 9FC0,9F,DD,7E,01,32,FF,9F,CD,748F  [8164]
133 DATA 9FC8,87,9F,CD,1B,BB,D8,C5,CD,7BAF  [2EBC]
134 DATA 9FD0,91,9F,C1,0B,78,B1,20,E3,7677  [0DAE]
135 DATA 9FD8,C9,21,FF,FF,11,40,9C,01,7C61  [D5EA]
136 DATA 9FE0,00,40,ED,B8,C9,21,40,9C,00F0  [D2A0]
137 DATA 9FE8,11,FF,FF,01,00,40,ED,B8,28D2  [FCFE]
138 DATA 9FF0,C9,20,E3,C9,21,FF,FF,11,7F6B  [6A36]
139 DATA 9FF8,40,9C,01,00,00,00,04,29,0701  [0EF0]
140 DATA A000,CD,BA,BB,CD,E7,BB,32,BD,56BD  [DA7A]
141 DATA A008,A0,CD,6C,A0,21,8F,01,22,67D4  [4284]
142 DATA A010,BE,A0,11,00,00,3E,07,32,75E4  [0224]
143 DATA A018,C0,A0,CD,7C,A0,0E,00,3A,5362  [C08E]
144 DATA A020,C0,A0,47,E5,D5,C5,CD,F0,4A66  [36D4]
145 DATA A028,BB,C1,D1,21,BD,A0,BE,E1,7305  [51DC]
146 DATA A030,37,20,01,A7,CB,11,2B,2B,1FB1  [2F66]
147 DATA A038,10,E9,CD,AF,A0,79,CD,A6,24C8  [B516]
148 DATA A040,A0,13,E5,21,7F,02,37,ED,4903  [9A58]
149 DATA A048,52,E1,38,05,2A,BE,A0,18,14E0  [7578]
150 DATA A050,CC,23,7C,B5,C8,2B,11,00,6CDE  [72C2]
151 DATA A058,00,22,BE,A0,3E,07,BD,20,15F6  [0880]
152 DATA A060,B9,7C,B4,20,B5,3E,04,32,526A  [3170]
153 DATA A068,C0,A0,18,AE,3E,1B,CD,A6,4140  [B4C6]
154 DATA A070,A0,3E,41,CD,A6,A0,3E,07,5CBB  [6CE0]
155 DATA A078,CD,A6,A0,C9,E5,3E,42,CD,5009  [C7F2]
156 DATA A080,1E,BB,E1,28,02,E1,C9,3E,3D58  [69CA]
157 DATA A088,0D,CD,A6,A0,3E,0A,CD,A6,2BE4  [D536]
158 DATA A090,A0,3E,1B,CD,A6,A0,3E,4C,57B0  [61EC]
159 DATA A098,CD,A6,A0,3E,7F,CD,A6,A0,59C0  [3936]
160 DATA A0A0,3E,02,CD,A6,A0,C9,CD,2E,0BD0  [3710]
161 DATA A0A8,BD,38,FB,CD,2B,BD,C9,3A,4134  [2B3C]
162 DATA A0B0,C0,A0,FE,07,C8,AF,CB,11,52CB  [B01A]
163 DATA A0B8,CB,11,CB,11,C9,00,07,00,7FF6  [DDAA]
164 DATA A0C0,04,19,12,01,BF,22,4D,41,03BB  [E64E]
165 DATA *ENDE*                             [87CC]
166 adr=&9EE0:zeile=104:MEMORY adr-1        [186E]
167 READ d$:IF d$="*ENDE*"THEN 178          [B6A6]
168 pr=0                                    [1F1C]
169 FOR i=1 TO 8                            [3672]
170 READ a$:a=VAL("&"+a$)                   [E43E]
171 POKE adr,a:adr=adr+1                    [E41A]
172 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [499A]
173 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                       [A8B2]
174 NEXT i                                  [2808]
175 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                          [8B94]
176 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                   [CC1C]
177 zeile=zeile+1:GOTO 167                  [AF76]
178 SAVE"ADWANDEL.BIN",B,&9EE0,&1E1         [194C]
179 PRINT d$:END                            [440E]
```

**Listing 2. Der DATA-Lader erzeugt die Binär-Datei**

# Als wär's ein IBM

**Mit dem Schneider PC bereichert ein weiterer IBM-kompatibler PC den Markt. Dieses Gerät ragt durch seinen außergewöhnlich niedrigen Preis aus der Masse der PC-Nachbauten heraus. Ob der Schneider PC auch leistungsmäßig Maßstäbe setzt, haben wir kritisch untersucht.**

Jede Computerzeitschrift, die etwas auf sich hält, berichtete in den letzten Wochen vom Schneider PC. Mit Lob wurde nicht gespart, doch eine kritische Auseinandersetzung mit dem Gerät blieb nahezu aus. Wir wollen nun einen sachlichen Überblick über die Fähigkeiten des Schneider PC ermöglichen, um Ihnen Hilfestellung bei einer Kaufentscheidung zu geben.

Als wichtigstes Argument für den Kauf des Schneider PC führt der Handel die IBM-Kompatibilität an. Diese gewährleistet, daß der Käufer das riesige Angebot an MS-DOS-Software nutzen kann. Um die Kompatibilität des Schneider PC zum Original zu überprüfen, ist es notwendig, die Hardware beider Computer zu vergleichen.

Ein wichtiges Kriterium für die IBM-Kompatibilität ist der Festwertspeicher (ROM), der die Systemparameter des Computers enthält. Das ROM des Schneider PC darf aus urheberrechtlichen Gründen nicht mit dem ROM des IBM-PC identisch sein. Außerdem ist das Innenleben des IBM-ROM ein wohlgehütetes Geheimnis. BasicA, das speziell für den IBM-PC geschrieben wurde und einzelne Maschinensprache-Routinen des IBM-ROM verwendet, ist deshalb auch nicht auf dem Schneider PC lauffähig. Ein Trost: Auf vielen anderen PC-Nachbauten läuft es ebensowenig.

Die Funktionen, die für das Betriebssystem MS-DOS (bei IBM heißt es PC-DOS) wichtig sind, wie zum Beispiel das Booten von MS-DOS, führt das Schneider-ROM einwandfrei aus, so daß hier kein Unterschied zum ROM des IBM-PC zu bemerken ist. Nur Computerspiele (speziell Actionspiele), die mit hochauflösender Grafik arbeiten und aus Geschwindigkeitsgründen direkt auf Routinen des ROM zugreifen, sind oft nicht lauffähig.

Der IBM-PC arbeitet mit dem Mikroprozessor 8088 von Intel. Dieser Prozessor verarbeitet intern 16 Bit parallel, verfügt extern jedoch nur über einen 8 Bit breiten Datenbus. Deshalb muß der 8088 Daten im Multiplex-Betrieb übertragen und übernehmen, das heißt, jedes 16-Bit-Wort wird in 2 Byte aufgeteilt, die nacheinander über den Datenbus laufen. Dieses Verfahren kostet den Prozessor wertvolle Arbeitszeit.

Der Schneider PC hat im Gegensatz zum IBM-PC einen Mikroprozessor 8086 eingebaut. Dieser verfügt über einen 16 Bit breiten Datenbus. Der Prozessor überträgt oder übernimmt ein 16-Bit-Wort als komplettes Stück und erreicht dadurch eine höhere Verarbei-



tungsgeschwindigkeit. Trotzdem sind der 8088 und der 8086 zueinander kompatibel, da sie über einen identischen Befehlssatz verfügen. Bei passender Beschaltung des 8086 läuft auf diesem Prozessor die gleiche Software wie auf dem 8088.

Da der Mikroprozessor des IBM-PC nur mit 4,77 MHz getaktet ist, der Schneider PC hingegen mit 8 MHz läuft, ergibt sich eine weitere Geschwindigkeitssteigerung zugunsten des Schneider PC. Insgesamt hat die gegenüber dem IBM-PC veränderte Hardware des Schneider PC zur Folge, daß der Schneider PC gut doppelt so schnell wie sein Vorbild ist.

Dies kann unter Umständen auch zu Kompatibilitätsproblemen führen. Ein Programm, das für seinen Ablauf wichtige Zeiten nicht über die eingebaute Hardware-Uhr, sondern über Warteschleifen bestimmt, verliert beim Schneider PC das Timing. Das kann Fehlfunktionen auslösen. Auf die Zeitsteuerung über Warteschleifen greifen jedoch nur einige wenige Programme zurück, da auch andere IBM-Kompatible mit höheren Taktfrequenzen als 4,77 MHz arbeiten.

Störend wirkt sich die höhere Taktfrequenz in der Regel bei Geschicklichkeitsspielen aus, die durch die höhere Geschwindigkeit nicht selten unspielbar werden. Das Problem mit Spielen, die auf das ROM zugreifen, wurde bereits erwähnt.

Anwendungsprogramme ohne solche Tricks – und das sind die meisten – laufen jedoch einwandfrei auf dem Schneider PC (zum Beispiel Turbo Pascal, dBase, Word, Worstar, Lotus).

Neben dem Prozessor 8086 besitzt der Schneider PC zusätzlich einen freien Sockel für den Arithmetik-Prozessor 8087. Dieser Prozessor beschleunigt die Rechenzeit des 8086 erheblich und übt keinerlei störenden Einfluß auf die anderen Funktionen des Computersystems aus. Da der Baustein jedoch mehrere hundert Mark kostet, hätte sein Einbau den Kaufpreis des Computers entsprechend erhöht. Für Anwender, die häufig langwierige mathematische Operationen durchführen müssen, ist der Kauf des 8087 durchaus zu empfehlen. Sie müssen den Baustein lediglich bei ausgeschaltetem Computer in die Fassung stecken – fertig!

Der Schneider PC enthält serienmäßig 512 KByte RAM. Eine Erweiterung um zusätzlich 128 KByte auf 640 KByte ist problemlos und geht genauso einfach vor sich wie das Einsetzen des Arithmetik-Prozessors. Nur 18 Speicher-Bausteine müssen Sie in die dafür vorgesehenen leeren Sockel stecken. Wenn Sie den Speicher darüber hinaus vergrößern möchten, wird allerdings der Einbau einer Speicher-Erweiterungskarte notwendig.

Eine Farbgrafikkarte, die bei vielen IBM-Kompatiblen erst dazugekauft werden muß, ist in der Grundausstattung des Schneider PC bereits enthalten. Das Gerät kann mit der vorhandenen Hardware maximal 16 Farben in einer Auflösung von 640x200 Bildpunkten darstellen. Dies reicht zwar nicht an die Qualitäten einer EGA-Karte (der »gehobene Standard«) heran, ist jedoch mehr als für Kompatible üblich.

Auch eine parallele (Centronics) und eine serielle Schnittstelle (RS232C), die nicht unbedingt zur Standard-Ausrüstung eines PCs zählen, sind im Schneider PC eingebaut. Die integrierte batteriegepufferte Uhr, der Mausanschluß (mit Maus), die Joystickbuchse und der leistungsstarke Lautsprecher runden die elektronische Hardware des Computers sinnvoll ab.

Der Schneider PC kommt wahlweise mit einem oder zwei eingebauten 5¼-Zoll-Laufwerken ins Haus. Die Schneider-Laufwerke arbeiten ausschließlich im IBM-Format, so daß nur 360 KByte Speicherplatz pro Diskette zur Verfügung stehen. Eine Abweichung vom Standard durch erhöhte Speicherkapazität wäre in diesem Fall begrüßenswert gewesen.

Wer einen Schneider PC mit zwei Laufwerken besitzt und sein System um

**Sehr aufschlußreich: der Blick in den »Bauch« des neuen Schneider PC. Selbst die Farbgrafikkarte ist serienmäßig in der Grundausstattung vorhanden**

Schnittstellenbaustein
Quartz
Disketten-Controller
Custom-Chip
Timer-Baustein
Sockel für Arithmetik-Prozessor
3 Steckplätze für Erweiterungskarten
512 KByte-RAM
Steckplätze für 128 KByte-RAM
ROM
Mikroprozessor
Custom-Chip
Custom-Chip

eine Schneider-Festplatte (10 oder 20 MByte) erweitern möchte, muß ein Laufwerk wieder ausbauen und durch die Festplatte ersetzen. Besser ist in diesem Fall der Kauf einer Festplatten-Karte zum Einstecken, denn dadurch können Sie weiterhin beide Laufwerke ohne Einschränkungen benutzen.

Negativ anzumerken ist, daß das Netzteil des Schneider PC schwach dimensioniert ist. Schneider hat bereits reagiert und einen kleinen Umbau vorgenommen, damit das Gerät nicht zu leicht abstürzt. Doch die Leistungsfähigkeit des Netzteils verbessert sich dadurch nicht. Aus diesem Grund bietet Vortex für 298 Mark ein 100 Watt starkes Schaltnetzteil für den PC an; eine Ausgabe, die jedem vorsichtigen Benutzer zu empfehlen ist.

Schneider liefert zu seinem PC ein Softwarepaket, das zwei Betriebssysteme, eine Programmiersprache und ein Malprogramm umfaßt. Dadurch ist der Anwender vom Start weg mit Software reichlich ausgestattet. Hätte Schneider auch noch ein Textverarbeitungssystem beigefügt, wäre der Service vollkommen.

Mit MS-DOS 3.2 von Microsoft erhalten Sie das Standard-Betriebssystem für PCs. Dieses Betriebssystem eröffnet den Zugriff auf das gewaltige Angebot an MS-DOS-Software. Die Version 3.2 bietet zusätzlich den Vorteil, netzwerkfähig zu sein. Mehrere PCs, die mit MS-DOS 3.2 laufen, können untereinander verbunden werden und gemeinsam Programme und Daten verarbeiten.

DOS Plus 1.2 ist eine noch relativ neue Entwicklung von Digital Research. Auf den ersten Blick unterscheidet sich DOS Plus von MS-DOS durch eine Statuszeile am unteren Bildschirmrand. Das Betriebssystem ist voll kompatibel zu CP/M 86 und teilkompatibel zu MS-DOS. Alle CP/M-Fans dürfen sich freuen, denn auch der PIP-Befehl ist unter DOS Plus implementiert.

Zu den Vorteilen von DOS Plus zählt die Fähigkeit zum Multitasking, das heißt, das Betriebssystem arbeitet mehrere Programme gleichzeitig ab (MS-DOS soll diese Fähigkeit erst ab der Version 5.0 erhalten.) Näher betrachtet erweisen sich allerdings die Multitasking-Talente von DOS Plus als recht bescheiden und unkomfortabel.

Der Anwender hat die Möglichkeit, bis zu drei Programme im Hintergrund ablaufen zu lassen, während er sich im Vordergrund mit dem Hauptprogramm beschäftigt. Für die Hintergrund-Programme gelten jedoch mehrere Einschränkungen, die den Anwendungsbereich des Multitasking stark begrenzen:

- Interaktive Programme (das heißt Programme, die vom Benutzer Eingaben anfordern und Ergebnisse ausdrucken) sind nicht lauffähig.
- Der Zugriff aus einem Hintergrund-Programm in das Vordergrund-Programm ist nicht erlaubt.
- Nur Programme mit der Extension CMD können im Hintergrund ablaufen. Dies hat zur Folge, daß MS-DOS-Programme mit den Extensions COM und EXE immer im Vordergrund laufen müssen. Da unter DOS Plus nur zwei Anwendungsprogramme mit der Extension CMD geliefert werden (das Terminprogramm »Alarm« und der Drucker-Spooler »Print«), sind die Multitasking-Möglichkeiten ohne zugekaufte Software kaum nutzbar.

## Schwierigkeiten mit Multitasking

Wenn im Multitasking-Betrieb ausschließlich Programme mit der Extension CMD ablaufen, gibt es keine Probleme, da diese Programme Speicherplatz und Rechenzeit selbständig unter sich aufteilen. Sollen dagegen COM- oder EXE-Programme im Vordergrund arbeiten, muß der Anwender COM-Programmen mit dem Befehl COMSIZE einen Speicherbereich zuweisen, und EXE-Programmen mit ADDMEM zusätzlichen Speicherplatz für Programmdaten reservieren.

Wieviel Speicherplatz ein CMD-Programm im Hintergrund belegt, läßt sich mit dem Befehl BACKG abfragen. Den Speicherbedarf von COM- und EXE-Dateien muß man dagegen im Handbuch zu den Programmen nachschlagen. Die Angaben in den Unterlagen sind jedoch zum Teil sehr ungenau, und auch der Speicherbereich für Programmdaten ist zu berücksichtigen. So bleibt oft nichts anderes übrig, als den tatsächlichen Speicherbedarf durch Ausprobieren herauszufinden. Unter Umständen kann es passieren, daß man meint, den wahren Speicherbedarf eines Programmes ermittelt zu haben. Bis es dann eines Tages abstürzt, mit einem 20-KByte-Textfile im Speicher, ungesichert ...

Der Befehl SLICE bestimmt die Aufteilung der Rechenzeit zwischen den einzelnen Programmen. Auch hier ist wieder geduldiges Probieren nötig, um die geeigneten Werte für bestimmte Anwendungen herauszufinden.

Das interessanteste Produkt im Softwarepaket des Schneider PCs ist GEM 2.0 mit der grafischen Benutzeroberfläche GEM-Desktop und dem Malprogramm GEM-Paint. GEM bietet gegenüber den beiden Betriebssystemen den Vorteil einer problemlosen Einführung in das System durch grafische Symbole (Icons). Schon nach kurzer Zeit ist der Anwender in der Lage, einfache Arbeiten wie Formatieren, Kopieren und Löschen durchzuführen. Die Befehlseingabe erfolgt (fast) ausschließlich über die Maus. Dies verringert die Berührungsängste von Anfängern vor dem Computer ganz beträchtlich.

GEM-Desktop stellt neben den grundlegenden Funktionen eine Uhr mit Datum, einen Taschenrechner und einen Drucker-Spooler für das Ausdrucken »nebenher« bereit. Durch die Art der grafischen Darstellung belegt GEM im Vergleich zu den beiden Betriebssystemen viel Speicherplatz. Dabei ist das fenster- und menügeführte GEM in seinen Funktionen nicht so leistungsfähig wie die textorientierten Betriebssysteme.

Basic 2 ist das bislang leistungsstärkste Basic von Locomotive Software und stellt nach den Basic-Versionen für die CPCs und dem Basic für den Joyce einen neuen Höhepunkt auf dem Gebiet der Basic-Interpreter dar. Basic 2 verfügt über knapp 300 Befehle und kommt ohne Zeilennumerierung aus, da Sprünge und Unterprogrammaufrufe über Labels erfolgen. Die Unterprogramme gestatten die Verwendung von lokalen Variablen.

Die Verarbeitung von Labels und lokalen Variablen in Unterprogrammen erlaubt dem Basic-Programmierer, eine Modulbibliothek aus häufig benötigten Unterprogrammen anzulegen, die sich problemlos in neue Programme einbinden lassen. Wer mit dieser Methode bereits in Pascal arbeitete, weiß die Vorteile zu schätzen.

## Basic 2 kontra GW-Basic

Zwei Nachteile von Basic 2 wollen wir jedoch nicht verschweigen: Zum einen ist Basic 2 zu GW-Basic, das den Standard unter den PC-Basic-Dialekten darstellt, nicht voll kompatibel, und zum anderen läuft Basic 2 nur unter GEM, so daß der PC-Besitzer beide Programme besitzen muß, um unter Basic 2 geschriebene Programme nutzen zu können.

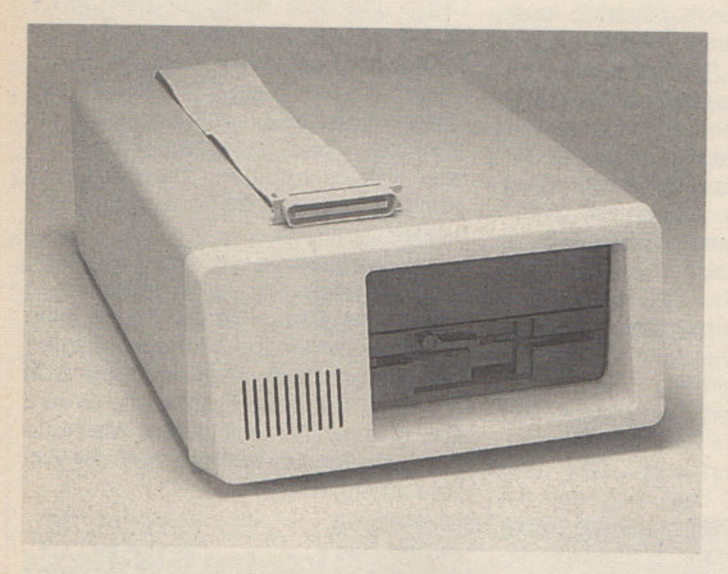
Unser kleiner Streifzug durch die Hard- und Software des Schneider PC vermittelt nur einen ersten Eindruck von den Fähigkeiten, die dieses Gerät seinem Anwender bietet. Nach einem Vergleich des Leistungsvermögens des Schneider PC mit dem seiner Konkurrenten gelangt man jedoch zu dem Schluß, daß der Schneider PC zur Zeit einer der preiswertesten und leistungsfähigsten IBM-Kompatiblen ist. Eine Kombination, die den Schneider PC auch für Heimanwender sehr attraktiv macht.

(ma)

# MS-DOS auf dem CPC

**Machen Sie aus Ihrem CPC einen PC. Die neuesten Informationen über zwei Emulatoren!**

Für Aufruhr sorgte vor kurzem der erste MS-DOS-Emulator für die Schneider-CPC-Familie. Genaugenommen handelt es sich dabei eher um einen fast vollständigen, IBM-kompatiblen Computer ohne Tastatur und Bildschirm. Die Funktionen der beiden letztgenannten Baugruppen übernimmt bei diesem PC-»Kern« des deutschen Herstellers Kersten & Partner eben ein CPC. Der Emulator besitzt ein eigenes Laufwerk im IBM-Format. Ein Gerät stand uns bereits zum Test zur Verfügung. Es vereint einen 8088-Prozessor (mit 5 Megahertz getaktet), 512 KByte Arbeitsspeicher, ein 5$^{1}/_{4}$-Zoll-Diskettenlaufwerk im IBM-Standardformat und einen langen Steckplatz für Erweiterungen in sich. Die Kopplung an den CPC erfolgte mit einem Flachkabel zum Erweiterungsport. Der RSX-Befehl »PC« sorgte dann für Aufregung in der Redaktion. Keiner konnte zunächst glauben, was er dort sah: Wohlbekannte MS-DOS-Software lief ohne jegliche Mühen auf dem »8-Bit-Computer« CPC 464. Selbst von der Geschwindigkeit her gab es nichts an der neuen Kombination auszusetzen. Sogar eine IBM-Grafikkarte emuliert sie auf dem angeschlossenen CPC. Damit ist die Lauffähigkeit der meisten Programmpakete



**Der erste MS-DOS-Emulator für CPCs**

## Preiswerter Nachfolger

sichergestellt. Aber zwei Kritikpunkte gab es dann doch. Der Lieferumfang umfaßt kein Betriebssystem. Man ist also darauf angewiesen, irgendwo zufällig eine Diskette zu »finden«, die das MS-DOS enthält. Während man mit diesem Umstand noch leben kann, wiegt der zweite um so schwerer. Mit einem Laufwerk kostete der Zusatz immerhin 1645 Mark. Also haben sich die Entwickler besonnen und inzwischen einen Nachfolger konzipiert, der sich in dieser Beziehung erheblich verbraucherfreundlicher gibt. Leider war er jedoch zum Zeitpunkt der Drucklegung noch nicht fertiggestellt. Deshalb erhalten Sie hier Vorab-Informationen, die uns der Hersteller bekanntgab. Mit seinem V20-Prozessor (8088-kompatibel) verwaltet der neue Emulator seinen eingebauten Arbeitsspeicher von 256 KByte. Das Diskettenlaufwerk läßt sich vom CPC aus als normales Zweitlaufwerk benutzen. Gegen zirka 100 Mark läßt sich der Speicher auf 512 KByte aufstocken. Ein zweites Laufwerk gibt es für etwa 300 Mark. Für schnelle Berechnungen läßt sich ein 8087-Coprozessor nachrüsten (zirka 450 Mark). Der Preis des Grundgerätes des Emulators ist mit 1098 Mark im Preis drastisch gefallen. Die Erstauslieferung wird wahrscheinlich im Januar beginnen, eine Joyce-Version soll im Februar folgen.

## Die Konkurrenz schläft nicht

Jetzt wirklich kurz vor der Fertigstellung steht der Emulator der Firma Vortex. In einem Interview konnten wir auch hier schon im voraus die wichtigsten Daten erfahren. Das Konzept der Vortex-Entwickler weicht vom vorher beschriebenen weitgehend ab. Zwar arbeitet dieser Emulator auch mit einem 8088-Prozessor, aber hier ist sein Arbeitstakt von 4,77 auf 8 Megahertz umschaltbar und somit eine höhere Geschwindigkeit zu erzielen. Auf der Platine wurde ein Sockel freigelassen, der im Bedarfsfall den Arithmetik-Coprozessor 8087 aufnimmt. Im Lieferumfang ist das Betriebssystem (MS-DOS 3.2) und der Interpreter GW-Basic mit kompletter deutscher Dokumentation, aber kein Diskettenlaufwerk, enthalten. Statt dessen beabsichtigt Vortex, die 5$^{1}/_{4}$-Zoll-Laufwerke aus dem eigenen Haus zu nutzen. Wer also bereits im Besitz einer solchen F1- oder M1-Station ist, benötigt keine weitere Peripherie. Den anderen Käufern stellt sich zur Anschaffung einer solchen Konfiguration als Alternative eine IBM-kompatible Controllerkarte nebst passendem Laufwerk. Mit dem Schneider-Monitor steht ein Textbildschirm zur Verfügung. Wer viel mit Grafik arbeitet, kauft sich eine entsprechende Erweiterungskarte und eventuell einen dazu passenden Monitor. Durch die Emulation der IBM-Laufwerke kann es bei einigen Programmen aber zu Kompatibilitäts-Problemen kommen, wenn deren Kopierschutz tief in den (hier ja nicht vorhandenen) IBM-Diskettencontroller eingreift. Auch in diesem Fall hilft als Hardwarelösung der Kauf einer Controllerkarte mit passendem Laufwerk. Von diesen zusätzlichen Karten finden in der Grundausführung drei Stück Platz. Die Zahl der Steckplätze läßt sich durch Anstecken weiterer Platinen erhöhen. Die beiden beschriebenen Karten, ein passender Monitor und eine IBM-Tastatur verschaffen Ihnen einen vollwertigen, unabhängig vom CPC arbeitenden, IBM-kompatiblen PC.

Das Grundgerüst besteht aus einer vierteiligen Steckplatzerweiterung zum CPC. Sie steckt in einem Gehäuse, das unter dem Monitor Platz findet und beherbergt bereits eine Steckkarte: den Emulator. Bleibt abzuwarten, was auf diesem Stecksystem basierend in Zukunft noch an Erweiterungen auf uns CPC-Besitzer zukommt. Denkbar ist in dieser Richtung alles; vom IBM-AT bis zum 68000er-Computer. Interessant ist das System allemal. Der Preis für die Grundversion liegt voraussichtlich bei 858 Mark. (ja)

Kersten & Partner, Wildbacher Mühle 83, 5100 Aachen, Telefon 0241/171067
Vortex Computersysteme, Falterstraße 1-5, 7101 Flein, Telefon 07131/52061

# Das Horten Computer-Center.
# Mehr als Sie erwarten – für weniger als Sie denken!

Im Computer-Center von Horten finden Sie genau das Zubehör, mit dem es sich noch leichter und sicherer, vielseitiger und erfolgreicher arbeiten läßt. In einer Qualität, hinter der Horten steht!

Computer
Center

**Wir bieten Ihnen ein umfangreiches Sortiment an:**
- Farbbänder für alle gängigen Drucker
- Verbindungskabel Computer/Peripherie
- Peripheriegeräte für Commodore, Atari, Schneider
- Großes Software-Angebot
- Reichhaltige Auswahl an Joysticks
- Breites Literatur-Sortiment
- Druckerpapier und Computer-Etiketten
- Marken-Disketten zu besonders günstigen Preisen.

Zum Beispiel:

**DISK-MASTER**
**Wendediskette 5.25"**, beidseitig nutzbar, mit 2 Schreibschutzkerben – 2 Indexlöchern
**10-Stück**-Packung nur **19.95**

**DISK-MASTER**
**Color-Diskette 5.25"**, die Wendediskette in 5 versch. Farben zur besseren Archivierung Ihrer Daten
**10-Stück**-Packung nur **24.95**

**DISK-MASTER**
**2D-Diskette** mit 96 tpi, 5.25", für hochwertige Diskettenlaufwerke
**10-Stück**-Packung nur **59.–**

**Sollte kein Horten-Haus in Ihrer Nähe sein, nutzen Sie bitte diesen**

## Bestellcoupon

An Horten AG, EA 634 · Am Seestern 1, 4000 Düsseldorf 11

| Ich bestelle hiermit | | Stück |
|---|---|---|
| Disk-Master Wendediskette 5.25" 10er-Pckg. | 19.95 | |
| Disk-Master Color-Diskette 5.25" 10er-Pckg. | 24.95 | |
| Disk-Master 2D-Diskette 5.25" 10er-Pckg. | 59.– | |

Wir liefern verpackungs- und versandkostenfrei in der Bundesrepublik. Sie zahlen lediglich die Zustell- und Rücküberweisungsgebühr.

Name ____ Vorname ____ Alter ____

Straße, Nr. ____ PLZ ____ Ort ____

Datum ____ Unterschrift ____




Horten
Horten
Horten

Computer-Center finden Sie in: Aachen · Baden-Baden · Berlin, Märkisches Viertel · Bielefeld · Braunschweig · Bremen · Bremerhaven · Dortmund · Düsseldorf, Berliner Allee · Duisburg · Essen · Gevelsberg · Gießen · Hagen · Hamburg, Mönckebergstraße · Hamburg-Wandsbek · Hannover · Hamm · Heidelberg · Heidenheim · Heilbronn · Hildesheim · Ingolstadt · Kempten · Kiel · -Zentrum · Mannheim · Moers · Münster · Neuss · Nürnberg · Oldenburg · Osnabrück · Pforzheim · Pirmasens · Regensburg · Reutlingen · Schweinfurt · Stuttgart · Trier · Ulm · Viersen · Wattenscheid

Duisburg - Hamborn
Krefeld · Ludwigshafen
Witten · Worms

# MS-DOS für Umsteiger

**Der Befehlsaufbau des Betriebssystems MS-DOS ähnelt stark dem älteren CP/M. Wer die Struktur von CP/M kennt, dem wird der Umstieg leichtfallen.**

In diesem Einführungskurs zeigen wir Ihnen den Aufbau und die Befehle von MS-DOS.

Vom Aufbau und der Bedienung her sind sich MS-DOS und CP/M sehr ähnlich. Das beginnt bereits beim Einschalten des Computers. Hierbei wird MS-DOS automatisch geladen (man spricht in diesem Zusammenhang auch vom »Booten«) und der Computer meldet sich nach kurzer Zeit mit dem Prompt-Zeichen, das sozusagen die Frage des Computers »Na, was jetzt?« an den Anwender darstellt. Das Prompt ist zugleich die Laufwerkskennung.

In der Regel wird unter MS-DOS mit mehreren Laufwerken gearbeitet, die durch Buchstaben gekennzeichnet sind. Beim IBM-PC und den meisten Kompatiblen – so auch beim Schneider PC – ist dem linken Floppylaufwerk der Buchstabe A, dem rechten (oder dem unteren) das B zugeordnet. Die Umschaltung zwischen den Laufwerken erfolgt durch Eingabe des Buchstabens und eines Doppelpunktes. Das Prompt ändert sich dann entsprechend.

Eine Eigentümlichkeit des DOS sind die Korrekturmöglichkeiten. Haben Sie bei der Eingabe einen Fehler gemacht, so steht Ihnen zunächst als Hilfe nur die Backspace-Taste (Linkspfeil) zur Verfügung. Dabei wird jeweils das links vom Cursor stehende Zeichen gelöscht. Sind Sie auf diese Weise zur fehlerhaften Stelle vorgedrungen und haben sie korrigiert, so müssen Sie die zuvor gelöschten Zeichen richtig eintippen. Bemerken Sie den Fehler erst nachdem Sie <RETURN> gedrückt haben (und das Betriebssystem dann eine entsprechende Fehlermeldung ausgegeben hat), so gibt es keine Möglichkeit mehr, mit dem Cursor in die weiter oben stehende fehlerhafte Zeile zu gelangen, um diese zu korrigieren. Eine gewisse Hilfestellung bieten Ihnen hierbei allerdings die Funktionstasten F1 und F3. Durch Drücken der <RETURN>-Taste wird nämlich die gesamte Eingabezeile in einen Zwischenspeicher übernommen. In der neuen Zeile können Sie dann mit der F1-Taste die zwischengespeicherte Zeile Zeichen für Zeichen wieder sichtbar machen und sie entsprechend korrigieren. Hier nun ein Beispiel: Angenommen, Sie wollen den Befehl »DISKCOPY A: B:« eingeben und tippen statt dessen aber »DISKCPY A: B:« ein. MS-DOS kann dieses Kommando jedoch nicht interpretieren und reagiert mit einer Fehlermeldung. Zur Korrektur drücken Sie nun fünfmal die <F1>-Taste, womit »DISKC« in der Eingabezeile steht. Hier muß nun das »O« (von COPY) eingefügt werden. Dazu drükken Sie die Insert-(<INS>-)Taste und geben den fehlenden Buchstaben ein. Jetzt steht »DISKCO« in der Eingabezeile. Den Rest des Befehls, der immer noch richtig zwischengespeichert ist, machen Sie wieder Zeichen für Zeichen mit der F1-Taste sichtbar. Eine weitere Vereinfachung bringt hier noch die F3-Taste. Sie bringt den Inhalt des Zwischenspeichers auf einmal auf den Bildschirm. Drücken Sie jetzt statt achtmal die <F1>-Taste einfach <F3>, so erscheint der Rest der Zeile.

## Umständliche Korrektur

Dies alles klingt zugegebenermaßen recht kompliziert und umständlich, erweist sich aber nach kurzer Einarbeitung als recht komfortabel.

Worin liegt nun der eigentliche Sinn dieses Betriebssystems? Nun, MS-DOS kontrolliert unter anderem alle Disketten- und Festplatten-orientierten Vorgänge. Dies beginnt beim Auflisten des Directory (das Inhaltsverzeichnis einer Diskette oder Festplatte) oder dem Formatieren von Disketten und reicht bis zu komplizierten Vorgängen wie zum Beispiel dem Sichern einer Festplatte. Diese Operationen steuern über 50 Befehle. Eine Übersicht finden Sie in der Tabelle des 5. Schneider-Sonderhefts auf Seite 18.

Man unterscheidet beim MS-DOS grundsätzlich zwischen zwei Befehlsarten. Da gibt es zum einen die sogenannten internen (residenten) und zum anderen die externen (transienten) Befehle. Als Grundgedanke gilt dabei, daß man die wichtigen Befehle (wie zum Beispiel DIR, COPY und so weiter) im Speicher des PCs hält, um sie nicht jedesmal, wie beim CP/M, von Diskette nachladen zu müssen.

Seltenere Kommandos wie zum Beispiel zum Formatieren oder zum Duplizieren eines Datenträgers, finden sich als kleine Routinen auf der MS-DOS-Diskette. Jeder Benutzer kann sich dann die externen Befehle, die er öfter benötigt, auf andere Disketten kopieren. Bild 1 zeigt das Verzeichnis solch einer typischen Diskette. Der Directory-Eintrag jeder Datei setzt sich aus mehreren Komponenten zusammen. Da ist zum einen der Name der Datei oder des Programms, der aus bis zu acht Zeichen besteht. Dem Namen angegliedert ist noch die sogenannte Extension (bis zu drei Buchstaben), die den Dateityp kenntlich macht. Ein Programm- oder ein Dateiname besteht also immer aus zwei Teilen, die durch einen Punkt getrennt sind.

So zeigt beispielsweise die Extension ».COM« an, daß es sich bei dieser Datei um einen Befehl (COMmand) handelt. Dies können, wie in unserem Beispiel (Bild 1), entweder externe Befehle des DOS oder normale eigenständige Programme sein. Beide behandelt das MS-DOS organisatorisch gleich. Programme können aber auch mit der Extension ».EXE« (EXEcutable File) im Diskettenverzeichnis stehen. Der Unterschied zwischen beiden Endungen liegt in der Art, wie das DOS die Programme in den Speicher lädt, was für den normalen Benutzer allerdings nebensächlich ist. Eine weitere wichtige Endung ist ».BAT« (BATch File, also eine Datei, die Befehlsfolgen enthält, nach denen der Computer ganze Arbeitsabläufe steuert, die aber selbst kein eigenständiges Programm darstellt).

Dateien mit diesen drei Extensions werden einfach durch deren Namen aufgerufen. Geben Sie zum Beispiel FORMAT ein, so startet das Betriebssystem das Formatierungsprogramm, wobei die Angabe der Extension (.COM) entfallen kann.

Gerade bei DOS-Befehlen kommt es oft vor, daß bestimmte Parameter (zum Beispiel Laufwerksangaben oder Dateinamen) mit dem Befehl angegeben werden müssen. Diese werden dann, durch einen Leerschritt getrennt, an den Befehl angehängt. Ein gutes Beispiel ist der Befehl »COPY«, mit dem Sie Dateien kopieren. Eine komplette Eingabe sieht beispielsweise so aus: »COPY A:BEISP.COM B:«. Hier wird das Programm »BEISP« mit der Extension ».COM« von Laufwerk A auf Laufwerk B kopiert. Die Laufwerke werden dabei durch den entsprechenden Buchstaben zusammen mit einem Doppelpunkt angegeben.

Die drei oben beschriebenen Dateinamen-Erweiterungen und die Extension ».SYS« (auf die hier nicht näher eingegangen werden soll), sind die einzigen, die das DOS für eigene Zwecke reserviert hält. Andere Endungen, wie

```
dir

 Diskette/Platte, Laufwerk A, hat den
Namen BEISPIEL
Verzeichnis von A:\

CAD           <DIR>      10.11.86  18.23
DATEN         <DIR>      10.11.86  18.23
KALK          <DIR>      10.11.86  18.23
COMMAND  COM     24380   28.05.86  12.00
DISKCOPY COM      6346   28.05.86  12.00
EDLIN    COM      7639   28.05.86  12.00
FORMAT   COM     11474   28.05.86  12.00
KEYBGR   COM      3278   28.05.86  12.00
CONFIG   SYS        36    1.10.86  23.12
AUTOEXEC BAT       135   10.11.86  18.25
     10 Datei(en)    301056 Byte frei
```

**Bild 1. Das Directory einer MS-DOS-Diskette enthält viele wichtige Informationen**

```
ECHO OFF
CLS
KEYBGR
WTDATIM
ECHO Bitte wählen Sie:
ECHO -----------------
ECHO 1. Textverarbeitung
ECHO 2. TabellenKalkulation
```

**Bild 2. Dieses AUTOEXEC.BAT-File sorgt für das automatische Laden eines Programms nach dem Einschalten**

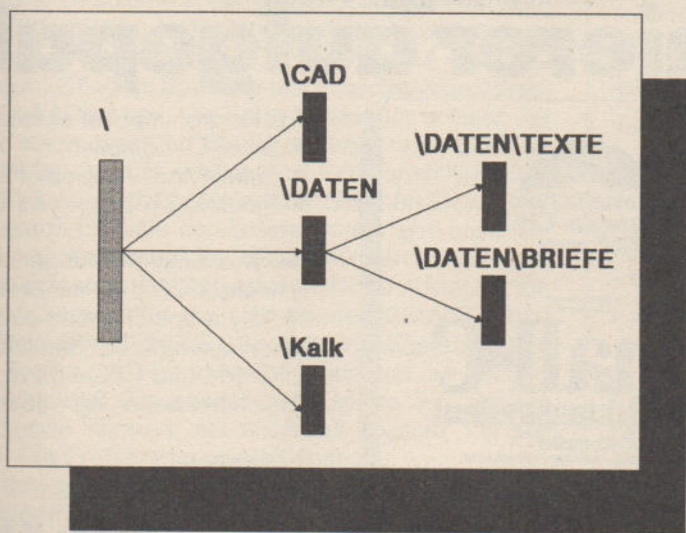



**Bild 3. Die Baumstruktur einer Beispieldiskette**

zum Beispiel ».DOC« und ».BAK«, können frei vergeben werden. Hiervon machen viele Anwendungsprogramme regen Gebrauch. So erkennen Sie in den meisten Fällen schon anhand der Extension, welches Programm eine Datei anlegte. Dabei haben sich bestimmte Endungen »eingebürgert«, zum Beispiel »TXT« und »DOC« für Textdateien, »BAK« und »SIK« für Backup-Dateien, »PRN« für Drucker- und »PIC« für Bilddateien.

Neben dem Dateinamen speichert MS-DOS bei jeder Änderung einer Datei das aktuelle Datum und die Zeit im Verzeichnis. Dies bringt besonders für Programmierer den Vorteil, daß Sie anhand des Directory feststellen können, welches die aktuellste Programmversion ist. Das setzt allerdings meist voraus, daß jeweils die aktuelle Zeit und das Datum nach jedem Einschalten oder jedem Reset eingegeben wird. Der Schneider PC gehört zu den wenigen Kompatiblen, bei dem diese umständliche Prozedur entfällt: Er verfügt über eine batteriegepufferte Uhr, die Zeit und Datum auch nach dem Abschalten des Computers speichert.

## Bequemer Autostart

Einen besonderen Status besitzen, wie schon erwähnt, in MS-DOS die sogenannten Batch-Dateien mit der Endung ».BAT«. Batchprocessing, zu Deutsch also Stapelverarbeitung, erlaubt – in gewissen Grenzen – eine Art Programmierung des DOS. Unter Programmieren ist hierbei aber mehr das Festlegen gewisser Abläufe zu verstehen. Ein solches Beispiel zeigt Bild 2. Dort ist ein Batch-File »AUTOEXEC. BAT« aufgelistet. Dieser Dateiname ist bei MS-DOS fest reserviert; der dort beschriebene Ablauf wird direkt nach dem Laden des Betriebssystems ausgeführt. Es empfiehlt sich jedem MS-DOS-Benutzer, von dieser Möglichkeit regen Gebrauch zu machen.

Das in Bild 2 aufgelistete Batch-Programm baut ein kleines Auswahlmenü für zwei Anwendungsprogramme auf. Zunächst zum Dateiaufbau: Als erster Befehl steht »ECHO OFF« in der Stapeldatei. Normalerweise werden alle Kommandos des Stapelprogramms vor ihrer Ausführung auf dem Bildschirm ausgegeben. »ECHO OFF« schaltet diesen Vorgang ab, so daß die Verarbeitung ohne Bildschirmecho geschieht. »ECHO« erfüllt aber auch noch die Funktion eines Ausgabebefehls. So erscheint zum Beispiel mit »ECHO Guten Tag« diese Nachricht auf dem Bildschirm. Der Befehl dient in unserem Beispiel dazu, das kleine Auswahlmenü aufzubauen. Möchten Sie dann zum Beispiel die Textverarbeitung anwählen, so drücken Sie einfach <1> und <RETURN>. Damit starten Sie ein weiteres Batch-Programm mit dem Dateinamen »1.BAT«. Dieses ruft dann seinerseits das Textverarbeitungsprogramm auf.

Neben dem »ECHO«-Befehl enthält die aufgezeigte Batch-Datei noch andere Befehle wie zum Beispiel »WTDATIM« und »KEYBGR«. Von besonderem Interesse für alle deutschen Computerbesitzer ist hierbei die Tastaturanpassung »KEYBGR«.

Das im PC eingebaute ROM enthält neben dem sogenannten BIOS – das

sind die für den Computer »lebensnotwendigen« Unterprogramme, die er nach dem Einschalten durchläuft – auch einen festgelegten Tastaturzeichensatz. Die gespeicherte Tastaturbelegung entspricht allerdings der amerikanischen Norm und stimmt daher nicht mit der zum PC gelieferten deutschen Tastatur überein. Das Programm »KEYBGR.COM« übernimmt die Tastaturanpassung.

Eine weitere, sehr sinnvolle Einrichtung bei MS-DOS sind die sogenannten Subdirectories (Dateizugriffspfade). Wie der Name bereits sagt, handelt es sich hierbei um Verzeichnisse im Verzeichnis. Wie in unserem Beispiel (Bild 1) zu sehen, befinden sich auf der dargestellten Diskette zwei Unterverzeichnisse, die jeweils der Anhang »DIR« kennzeichnet. Sie enthalten ihrerseits wieder Programme, die Sie wie gewohnt aufrufen. Mit dem Befehl »CD (Name des Subdirectory)« schalten Sie in das gewünschte Unterverzeichnis. Geben Sie nun »DIR« ein, so erscheint eine ganz gewöhnliche Auflistung der Dateien. Lediglich durch die Punkte am Anfang des Directory erkennen Sie, daß es sich hierbei um ein Unterverzeichnis handelt.

Programmaufrufe und Befehlseingaben wirken immer nur auf das aktuelle Inhaltsverzeichnis. Ein Zugriff auf ein Programm in einem Subdirectory erfolgt, indem Sie mit »CD« in dieses Unterverzeichnis schalten und die Anwendung dann ganz normal mit ihrem Dateinamen aufrufen.

Wichtig sind die Subdirectories erst für die Besitzer einer Festplatte. Denn gerade wenn das DOS sehr viele Dateien verwalten muß (bei einer 20-MByte-Platte sind dies immerhin 400 bis 800), wirkt sich diese Organisationsform sehr positiv auf die Übersichtlichkeit des Inhaltsverzeichnisses aus. So »verschwinden« alle Programme mit ihren Hilfsdateien in den verschiedensten Unterverzeichnissen. Am besten machen Sie sich die Organisation am Beispiel eines Baums klar. Bild 3 zeigt einen solchen Baum zum in Bild 1 dargestellten Directory.

Bestimmte Befehle, wie zum Beispiel »COPY«, fordern unter Umständen auch die Angabe des jeweiligen Unterverzeichnisses. Dies geschieht mit dem Backslash (Rückwärtsquerstrich) »\«, den Sie auf der Tastatur über die Tastenfolge »CTRL+ALT+<« erreichen. Angenommen, Sie möchten eine Datei von der Diskette in Laufwerk A mit dem Namen »ARTIKEL.TXT« auf eine Festplatte in ein Unter-Unterverzeichnis kopieren. Der Befehl sieht dann wie folgt aus: »COPY A:ARTIKEL.TXT C:\DATEN\TEXTE*.*«.

Angelegt wird ein solches Unterverzeichnis mit dem »MD«-Befehl (»make directory«), wobei dem Befehlswort der entsprechende Name folgen muß. Zum Löschen verwendet man das »RD«-Kommando. Es muß allerdings sichergestellt sein, daß das gewünschte Verzeichnis keine Dateien mehr enthält. Ist dies nicht der Fall, so müssen Sie zuerst alle Daten mit dem Befehl »ERASE *.*« oder »DEL *.*« aus dem Verzeichnis löschen.

Die Sternchen im obigen Befehl sind sogenannte Joker. Sie geben dem DOS zu verstehen, daß ein Befehl alle Dateien in diesem Unterverzeichnis betrifft. In dem gezeigten Beispiel wird das komplette Verzeichnis gelöscht. Auch beim »COPY«-Befehl finden die Joker Anwendung. So läßt sich zum Beispiel eine ganze Dateigruppe (mit einer bestimmten Extension) auf einmal spezifizieren. »COPY A:*.BAT B:« kopiert alle Dateien mit der Endung ».BAT« von Laufwerk A nach Laufwerk B.

Den kompletten Befehlssatz mit einer Kurzbeschreibung zu jedem internen beziehungsweise externen Befehl liefert Tabelle 1 im 5. Schneider-Sonderheft auf Seite 18.

(Christoph Sauer/Matthias Rosin/hg)

# Noch mehr Eingabekomfort

**Der neue Checksummer für den Schneider CPC ist da! »Explora 2.0« macht die Eingabe von Programmen noch einfacher.**

Zuerst einmal Informationen für alle, die noch nicht wissen, was »Explora« ist: Wenn Sie dieses Programm gestartet und wieder gelöscht haben, überprüft der Computer automatisch Ihre Eingaben auf Richtigkeit. Sobald Sie die Eingabe einer Programmzeile abschließen, erscheint eine vierstellige Hexadezimalzahl in eckigen Klammern auf dem Bildschirm. Das im Heft abgedruckte Listing enthält ebenfalls solche Zahlen. Stimmen die Prüfsummen auf dem Bildschirm und im Heft überein, haben Sie die Zeile korrekt abgetippt. Gibt es Unterschiede zwischen den Werten, sollten Sie auf Fehlersuche gehen und die Zeile korrigieren. Das alles konnte »Explora 1.0« auch schon. Der Vorteil der neuen Version besteht darin, daß Sie jetzt größere Freiheit bei der Eingabe der Zeilen haben. So akzeptiert unser Prüfsummenprogramm die Basic-Schlüsselworte in Klein- oder Großbuchstaben (auch gemischt). »PRINT« läßt sich mit dem Fragezeichen abkürzen. »Explora 2.0« läßt zum Beispiel für die Zeile »100 PRINT« folgende Eingaben zu:

```
100 PRINT
100 print
100 ?
100 PrInT
```

Die Zeilen müssen also nicht mehr schon beim Eintippen so aussehen wie im Heft, sondern erst beim Auflisten. Außerdem werden Prüfsummen nur noch für Programmzeilen ausgegeben, nicht mehr – wie früher – auch bei Direktbefehlen. Vor der Zeilennummer stehende Leerzeichen, Line-Feeds und Tabulatorzeichen überliest Explora jetzt selbsttätig. Leerzeichen innerhalb der Zeile wertet es aber weiterhin. Sie verändern also die Prüfsumme. Explora erlaubt auch die Verwendung des EDIT-Befehls. AUTO ist jetzt ohne Einschränkungen zu benutzen – allerdings nur beim CPC 664 und CPC 6128. Explora 1.0 liegt im Speicher fest zwischen den Adressen A000 und A086 hex. Die neue Version verschiebt der Basic-Lader automatisch im Speicher direkt unter HIMEM. So ist SYMBOL AFTER einwandfrei funktionsfähig. Eine kleine Einschränkung gibt es aber doch: Löschen Sie keinesfalls Zeilen durch Eingabe der Zeilennummer und anschließendes Drücken der ENTER-Taste! Die Zeile wird nämlich gar nicht wirklich gelöscht, sondern erscheint als Duplikat der folgenden Zeile. Verwenden Sie statt dessen DELETE. Statt »20« schreiben Sie »DELETE 20«. Das Wichtigste nicht zu vergessen: Explora 2.0 ist aufwärtskompatibel zur Version 1.0. Das heißt, daß Sie sowohl mit Explora 2.0 frühere Listings abtippen können, als auch mit Explora 1.0 alle zukünftigen. Die Prüfsummen sind identisch.

Aber bei den gedruckten Listings hat sich einiges geändert. Die Neuerungen betreffen die Darstellung von Leer- und Sonderzeichen. Statt "{5 SPACE}" steht jetzt im Listing "<5>" für fünf Leerzeichen. Um dies eindeutig vom tatsächlichen Programmcode zu unterscheiden, erscheint der Text unterstrichen. Die Steuerzeichen hießen bisher beispielsweise »{CTRL A}«. Jetzt steht hier die übersichtlichere Form A. Finden Sie im Listing also einen unterstrichenen Buchstaben ohne Klammern, müssen Sie gleichzeitig die CTRL-Taste drücken. Manche Programmautoren bestehen leider immer noch darauf, auch die Grafikzeichen von 128 bis 255 in Programme aufzunehmen. Solche Symbole stehen zukünftig in Klammern und sind als ASCII-Wert mit vorangehendem »G« für »Grafikzeichen« dargestellt. Das Zeichen 223 hat dann im Listing die Form <G223>. Die Zeichen können nicht von der Tastatur aus direkt eingegeben werden. Simpler Trick: Ausgabe des Zeichens mit »PRINT CHR$(223)« und Übernahme mit dem Copy-Cursor.

Sämtliche Listings sind im ASCII-Zeichensatz gedruckt. Deutsche Sonderzeichen erscheinen daher im Druck als Klammern und andere amerikanische Zeichen. Verwenden Sie ruhig an Stelle dieser Zeichen die entsprechenden deutschen.

(Martin Kotulla/ja)

```
100 ' **************************          [DFCC]
110 ' *                        *          [FADA]
120 ' *      EXPLORA V2.0      *          [761E]
130 ' *                        *          [DCDE]
140 ' **************************          [C3D4]
150 '                                     [E1BA]
160 DEF FNlsb(x)=255 AND UNT(x)           [39E0]
170 DEF FNmsb(x)=255 AND INT(x/256)       [8864]
180 SYMBOL AFTER 256:MEMORY HIMEM-161     [948C]
190 start=HIMEM+1:SYMBOL AFTER 240        [2092]
200 FOR i=&A000 TO &A09D:READ a$:sum=sum
    +VAL("&"+a$):NEXT i                   [B2C8]
210 IF sum<>19814 THEN PRINT "DATA-Fehle
    r!":END                               [FCCE]
220 RESTORE:FOR i=start TO start+&9D:REA
    D a$                                  [608E]
230 POKE i,VAL("&"+a$):NEXT i             [24D2]
240 FOR i=1 TO 5:READ a:a=a+start         [AC2A]
250 wert=PEEK(a)+PEEK(a+1)*256-40960+sta
    rt                                    [2776]
260 POKE a,FNlsb(wert):POKE a+1,FNmsb(we
    rt):NEXT i                            [01B2]
270 IF PEEK(6)=&80 THEN ed=&BD3A:POKE &B
    F20,&A4                               [56A8]
280 IF PEEK(6)=&7B THEN ed=&BD5B:POKE &B
    F20,&8A:RESTORE 470                   [760C]
290 IF PEEK(6)=&91 THEN ed=&BD5E:POKE &B
    F20,&8A:RESTORE 490                   [16FA]
300 POKE &BF21,&AC:POKE &BF22,PEEK(ed)    [71DE]
310 POKE &BF23,PEEK(ed+1):POKE &BF24,PEE
    K(ed+2)                               [9984]
320 POKE ed,&C3:POKE ed+1,FNlsb(start):P
    OKE ed+2,FNmsb(start)                 [9AE6]
330 IF PEEK(6)=&80 THEN END               [6044]
340 FOR i=1 TO 7:READ a$,b$:a=VAL("&"+a$
    )+start:b=VAL("&"+b$)                 [3306]
350 POKE a,FNlsb(b):POKE a+1,FNmsb(b):NE
    XT i                                  [0332]
360 DATA CD,22,BF,F5,C5,D5,E5,2A,20,BF,C
    D,61,DD,B7,28,62                      [5BFC]
370 DATA E5,2A,20,BF,CD,88,A0,E1,30,58,C
    D,04,EE,CD,A3,E7                      [5EF2]
380 DATA CD,63,E1,ED,4B,20,BF,21,00,00,0
    A,5F,16,00,19,03                      [DBF6]
390 DATA FE,00,20,F6,DD,2A,20,BF,01,00,0
    0,DD,7E,00,5F,16                      [4D3E]
400 DATA 00,19,04,F5,A8,47,F1,09,DD,23,F
    E,00,20,ED,3E,0D                      [E53C]
410 DATA CD,5A,BB,3E,0A,CD,5A,BB,3E,5B,C
    D,5A,BB,7C,CD,77                      [259A]
420 DATA A0,7C,CD,7B,A0,7D,CD,77,A0,7D,C
    D,7B,A0,3E,5D,CD                      [014A]
430 DATA 5A,BB,E1,D1,C1,F1,C9,1F,1F,1F,1
    F,E6,0F,C6,30,FE                      [A10A]
440 DATA 3A,38,02,C6,07,C3,5A,BB,CD,61,D
    D,B7,37,C8,CD,04                      [64AC]
450 DATA EE,D0,7E,FE,20,20,01,23,CD,D2,E
    6,37,9F,C9                            [0C36]
460 DATA &15,&5F,&63,&67,&6B              [3A22]
470 DATA 0B,DE52,1B,EED4,1E,E869          [7B14]
480 DATA 21,E259,89,DE52,8F,EED4,99,E7AA
                                          [0586]
490 DATA 0B,DE4D,1B,EECF,1E,E864          [1F52]
500 DATA 21,E254,89,DE4D,8F,EECF,99,E7A5
                                          [249A]
510 END                                   [AA1A]
```

**Listing. »Explora« macht Eingabefehler fast unmöglich**

| Steckbrief | |
|---|---|
| Programm: | Explora 2.0 |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora 1.0 |
| Datenträger: | Kassette/Diskette |

# Selbst ist der Programmierer

**Software selber schreiben ist gar nicht so schwer, wie viele denken. Besonders das Locomotive-Basic des Schneiders reicht Ihnen hierbei hilfreich die Hand. Es ist leicht zu erlernen, dabei aber trotzdem mit Befehlen für jeden Zweck ausgestattet.**

Action ist angesagt. »Starstrike«, »Ghosts'n Goblins«, »Mission Elevator« – hinter jedem Namen steckt ein fesselndes Spiel. Doch jedes Programm kostet Geld. Viel mehr als zwei neue Abenteuer sind für 100 Mark nicht zu bekommen. Da ist sehr schnell Ebbe im Geldbeutel. Schreiben Sie Ihre Software doch einfach selbst. Man muß nicht in die tiefsten Tiefen komplizierter Assembler-Programmierung einsteigen, um tolle Spiele (und nicht nur die) zu schreiben. Das eingebaute Locomotive-Basic macht Ihren Schneider CPC zu einem idealen Programmierwerkzeug.

»Breakout« ist ein immer wiederkehrender beliebter Hit der Spieleszene. Mit einem Ball müssen Sie aus einer Mauer Steine herausschlagen. Dabei darf der Ball nur auf den Schläger treffen und nicht auf den Boden fallen.

Jedes Programm muß als erstes den Computer auf seine Aufgabe vorbereiten und dann das eigentliche Problem bearbeiten. Unterprogramme, die mit verschiedenen Programmteilen zusammenarbeiten, stellt man an das Ende.

In den Zeilen ab 1000 finden Sie die Definitionen verschiedener Variablen und Felder. Ab Zeile 2000 erfolgt der Bildschirmaufbau. Einzelne Bereiche fangen in unserem Programm immer mit einer REM-Zeile an. Die Zeichen in solch einer REM-Anweisung werden beim späteren Bearbeiten vom Computer nicht berücksichtigt. Ihr Inhalt dient der Information des Programmierers.

Als erstes bringen wir den Rand des Spielfelds auf den Bildschirm. Zeile 1040 legt den Modus auf 1 fest. Für Spiele ist Modus 1 ein geeigneter Kompromiß, da die Auflösung mit 40 Zeichen pro Zeile akzeptabel ist und vier Farben gleichzeitig dargestellt werden. Als Beigabe wird mit MODE auch der Bildschirm gelöscht. Alte Eingaben stören also nicht. Aus der Zeichensatztabelle des Handbuchs suchen wir uns die Zeichen 149, 150, 154 und 156 für den Rand heraus. Das erste Zeichen kommt gleich oben links in die Ecke. Die folgende Schleife (Zeilen 2080 bis 2100) wird 38mal bearbeitet und setzt dabei jedesmal das Zeichen 154.

Solch eine FOR-NEXT-Schleife ist einer der wichtigsten Basic-Befehle überhaupt. Zuerst wird der Hilfsvariablen »i« der Startwert (in unserem Programm die 2) zugeteilt. Am Ende der Schleife (das ist bei der Anweisung »NEXT i«) wird i um 1 erhöht, und das Programm arbeitet mit dem zweiten Schleifendurchgang weiter. Dieses Spiel geht so lange weiter, bis der vorgegebene Endwert (bei uns 39) erreicht ist. Erst dann arbeitet der Computer mit den dahinterstehenden Befehlen weiter. In Ihrem Handbuch sehen Sie, daß durch die Option STEP die Schrittweite von 1 variiert werden darf. So darf Zeile 2080 auch

```
FOR i=2 TO 78 STEP 2
```

lauten.

In der nächsten Schleife finden Sie einen neuen Befehl. Mit LOCATE wird ein bestimmter Platz auf dem Bildschirm angesprochen. Dazu ist der Schirm im Modus 1 in ein Raster mit 25 Zeilen zu je 40 Zeichen eingeteilt. Wie Sie eine einzelne Position berechnen, sehen Sie in Bild 2. Das Zeichen 149 in Zeile 2140 wird somit immer in die erste Spalte, allerdings mit steigender Zeilennummer, geschrieben. Die Zeilennummer ist dabei durch die Variable i der FOR-NEXT-Schleife gegeben. Die Anweisung »TAB(40)« in Zeile 2150 sagt dem Computer, daß er als nächstes die Spalte 40 in der gerade aktuellen Zeile beschreiben soll. TAB ist damit im Prinzip nichts anderes als der Tabulator einer Schreibmaschine.

Das »;« hinter jeder PRINT-Anweisung verhindert, daß bei der nächsten Ausgabe auf dem Bildschirm automatisch eine neue Zeile begonnen wird. Der GOTO-Befehl in Zeile 2170 läßt den Computer in einer Endlos-Schleife verweilen, damit wir uns das Bild anschauen können. Nun tippen Sie Listing 1 ein und schauen sich an, was der Computer macht. Die Zahlen in den eckigen Klammern gehören zu unserem Prüfsummenprogramm »Explora«. Es überwacht Ihre Eingabe auf Fehler. Die Beschreibung dazu finden Sie auf Seite 41 in diesem Heft.

Als nächstes bauen wir die Mauer (Listing 2). Um später immer wieder herauszufinden, wo ein Stein steht, definieren wir das Feld »feld$« (Zeile 1020). Solche Felder sind verschiedene Variablen mit dem gleichen Namen. Man spricht ein bestimmtes Element mit seiner Positionsnummer an. Ein zweidimensionales Feld ist mit einem Schachbrett zu vergleichen. So wie Sie eine bestimmte Position dadurch eindeutig bestimmen, daß Sie die Spalten- und Zeilennummern angeben, so machen Sie das auch in einem Feld. Felder dürfen aber theoretisch unendlich viele Dimensionen haben. Allerdings übersteigt dann die Vorstellung von solch einem Speicher für den Menschen die Verständnisgrenze. »feld$« enthält für jede Bildschirm-Position ein Element – ist also zweidimensional.

Als nächstes legen wir die Farben der Steine fest. Jedem Farbstift ordnen wir eine Farbe aus der Tabelle des Handbuchs zu.

Die Schleife aus Listing 2 malt in die Bildschirmzeilen 7 bis 13 jeweils einen Stein. Der LOCATE-Befehl hängt diesmal von zwei ineinander verschachtelten Schleifen ab. Dies ist der einfachste Weg, einen Bildschirmbereich – oder ein Feld – zu beschreiben. Das Zeichen 233 steht für unseren Mauerstein. Gleichzeitig setzen wir in das Feld an der Stelle eine 1 ein. Im späteren Spiel sieht man dann jederzeit, ob hier ein Stein steht oder nicht.

In Zeile 2190 wird mit PEN der

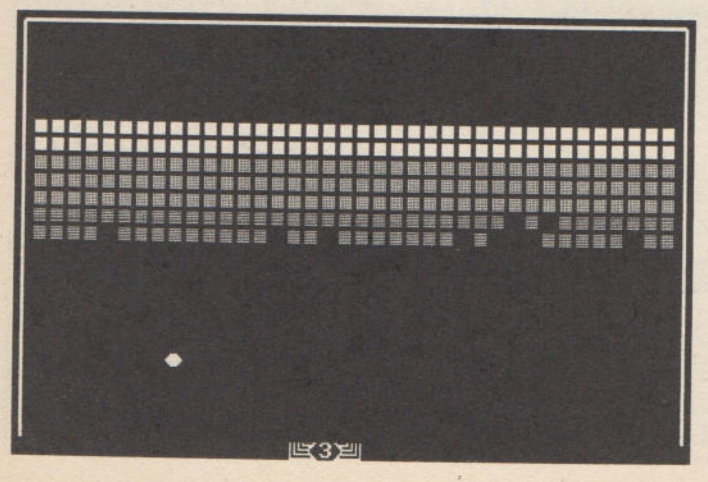


**Bild 1. »Breakout« – ein Actionspiel selbst programmiert**

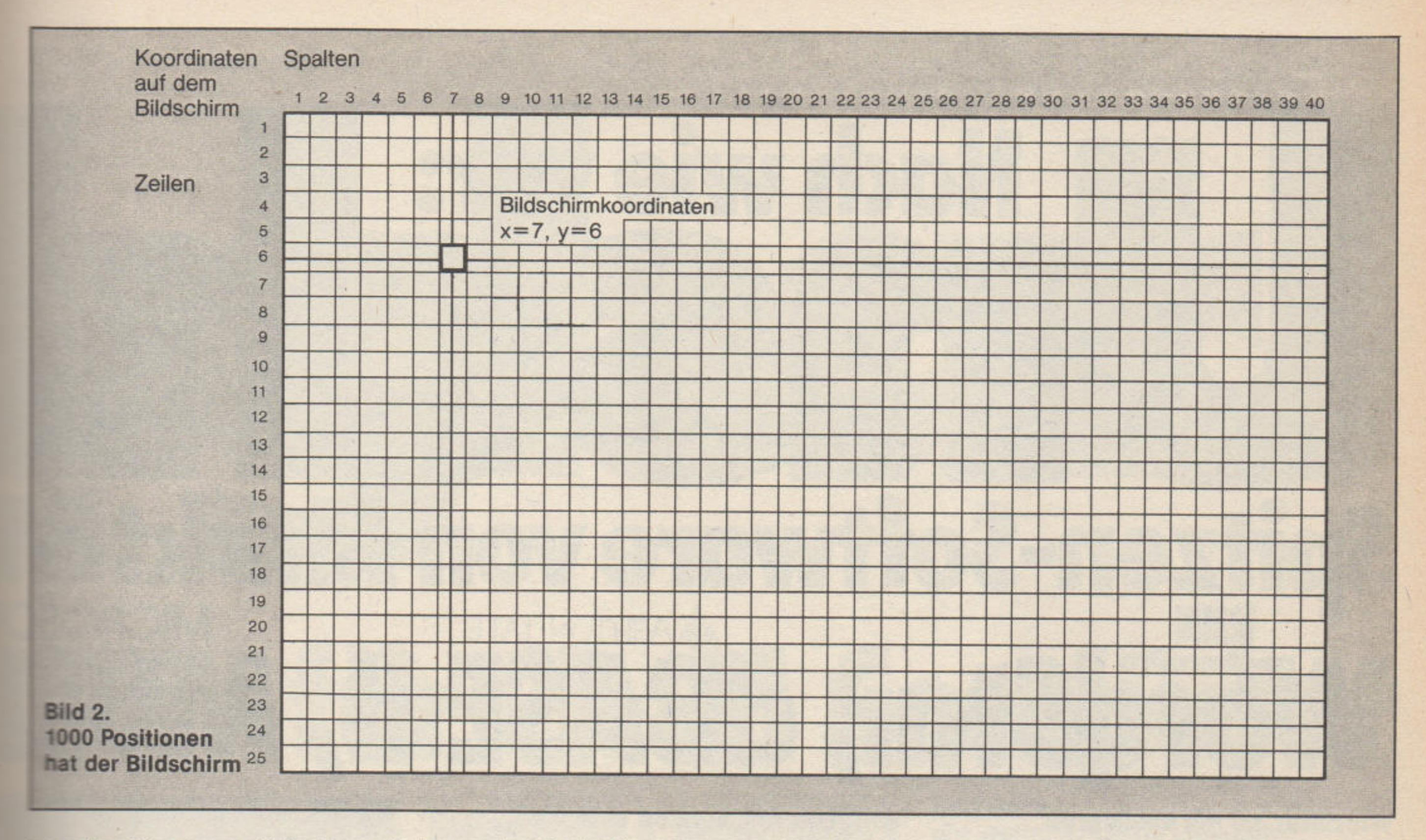



**Bild 2. 1000 Positionen hat der Bildschirm**

Schreibstift ausgewählt. Uns stehen die Ziffern 1 bis 3 (mit 0 ist ja der Hintergrund gezeichnet) zur Verfügung. Da i immer einen Wert von 7 bis 13 annimmt, liefert die Rechenanweisung »(i/3)–1« immer einen Wert zwischen 1,33 und 3,66. Mit INT wird davon nur der ganzzahlige Werte genommen – also genau unsere gesuchten Ziffern 1, 2 oder 3.

In Zeile 2260 wird noch der Stift auf 1 zurückgesetzt, und schon ist der Bildschirm fertig aufgebaut. Geben Sie jetzt Listing 2 ein. Die Anweisungen aus Listing 1 müssen aber im Speicher stehenbleiben. Das gilt übrigens für alle weiteren Programmteile. Alles zuvor Eingegebene muß im Speicher bleiben.

Als nächstes kommt unser Schläger an die Reihe. Bild 3 zeigt, wie er aussehen soll – 5 Symbole breit, und in der Mitte steht die Zahl des Balls. Zum Umdefinieren eines Zeichens dient der Befehl SYMBOL. Der erste Wert bezeichnet die Nummer des Zeichens. Er muß normalerweise größer als 239 sein. Es können damit ohne andere Anweisungen maximal 16 Zeichen neu definiert werden. Jedes Zeichen besteht aus 8 Byte. Diese geben verschlüsselt die einzelnen Zeilen eines Symbols wieder. Die Spalte ganz links hat dabei den Wert 128, die zweite 64, die dritte 32, die vierte 16, dann 8, 4, 2 und zum Schluß die Spalte ganz rechts den Wert 1. Die Zeilen 2020 bis 2050 definieren 4 Symbole. Falls Ihnen unser Muster nicht gefällt, so ist es leicht, einen anderen Schläger zu programmieren.

Zeile 4020 bringt den Schläger auf den Bildschirm. Dazu werden der Wert »ball« (Definition auf 1 in Zeile 1060) und die Position des Schlagers »posx« gebraucht. Alle Variablen, die von mehreren Programmteilen aufgerufen werden, legen wir in das Unterprogramm ab Zeile 9000 ab. Solch eine Routine wird mit GOSUB (siehe Zeile 1080) aufgerufen. Der Computer merkt sich, von wo der Aufruf erfolgt, und kehrt nach dem Ende der Unterroutine (das Kennzeichen dafür ist »RETURN«) zu dem Aufruf zurück. GOSUB-Routinen sind immer dann nützlich, wenn man bestimmte Anweisungen in den unterschiedlichsten Programmteilen braucht. »LOCATE posx,25« weist schließlich dem Schläger seinen Platz in der untersten Bildschirmzeile zu.

Im Hauptprogramm (Listing 4) wird zuerst nur der Schläger bewegt. Dieser muß immer in der Zeile 25 bleiben, darf aber zwischen den Spalten 1 und 34 hin und her fahren. 34 deshalb, weil der Schläger mit Leerfeldern 7 Zeichen lang ist. Die Zeilen 4010 und 4020 positionierten unseren Schläger korrekt. Für die Bewegung muß sich deshalb nur »posx« verändern.

INKEY heißt der Befehl, mit dem jede Taste des Computers abgefragt werden kann. Wird eine Taste gedrückt, so ist der Wert, der zurückgegeben wird, 0. Bei –1 ist die Taste nicht gedrückt. Tabelle 1 zeigt alle Werte, die man in den verschiedenen Kombinationen mit der SHIFT- und der CTRL-Taste bekommt.

Allen Tasten Ihres Schneiders ist dabei eine Nummer zugeordnet (siehe Handbuch). Die Cursortaste nach links hat die Ziffer 8 und die Cursortaste nach rechts die Ziffer 1. Zeile 4030 wertet die Tastaturabfrage aus. Die beiden Ausdrücke in den Klammern sind immer dann 0, wenn die Taste nicht gedrückt ist (1+(–1)). Bei gedrückter Taste wird die Abfrage 0 und der Ausdruck in der Klammer 1. Die 1 wird zu »posx« hinzugezählt beziehungsweise abgezogen und gleichzeitig der Variablen wieder zugeordnet. Zeile 4040 paßt auf, daß der Schläger nicht rechts oder links aus dem Bildschirm herausfährt.

In der booleschen Algebra ist ein Wert wahr (und damit gleich 1) wenn er richtig ist. Gilt also »posx=0« oder »posx=35«, so wird der Ausdruck in den Klammern »1«. Da der erlaubte Bereich für den Schläger zwischen 1 und 34 liegt, muß genau in diesem Augenblick eine 1 addiert (beziehungsweise abgezogen) werden. Zeile 4040 sorgt also für den richtigen Wertebereich. Der GOTO-Befehl ist dafür zuständig, daß das Hauptprogramm permanent bearbeitet wird.

Zu einem Spiel wird unser Programm erst, wenn auch der Ball eingebaut ist. Dazu benutzen wir eine Besonderheit des Schneider-Basics – den Interrupt. Zuerst definieren wir in Zeile 1050 die Spielstufe. Davon hängt die Geschwindigkeit des Balls ab. In den Zeilen 9010 bis 9040 stehen die Startposition und die Verschiebung. Zuerst soll dieser nach unten rechts fallen. Wir müssen also einfach den Zeilen- und den Spaltenwert um eins erhöhen.

Das Unterprogramm in Zeile 3010

# Nach wie vor:

# Unschl

# Spitzen-Software von Markt&Te

MicroPro. ASHTON·TATE MICROSOF

# WordStar, dBASE II, MULT

**WordStar 3.0 mit MailMerge**
Ein Bestseller unter den Textverarbeitungsprogrammen, der Ihnen bildschirmorientierte Formatierung, deutschen Zeichensatz und DIN-Tastatur sowie integrierte Hilfstexte bietet. Mit MailMerge können Sie Serienbriefe mit persönlicher Anrede an eine beliebige Anzahl von Adressen schreiben und auch die Adreßaufkleber drucken.

**dBASE II, Version 2.41**
dBASE II, das meistverkaufte Programm unter den Datenbanksystemen, eröffnet Ihnen optimale Möglichkeiten der Daten- und Dateihandhabung. Einfach und schnell können Datenstrukturen definiert, benutzt und geändert werden. Der Datenzugriff erfolgt sequentiell oder nach frei wählbaren Kriterien, die integrierte Kommandosprache ermöglicht den Aufbau kompletter Anwendungen wie Finanzbuchhaltung, Lagerverwaltung, Betriebsabrechnung usw.

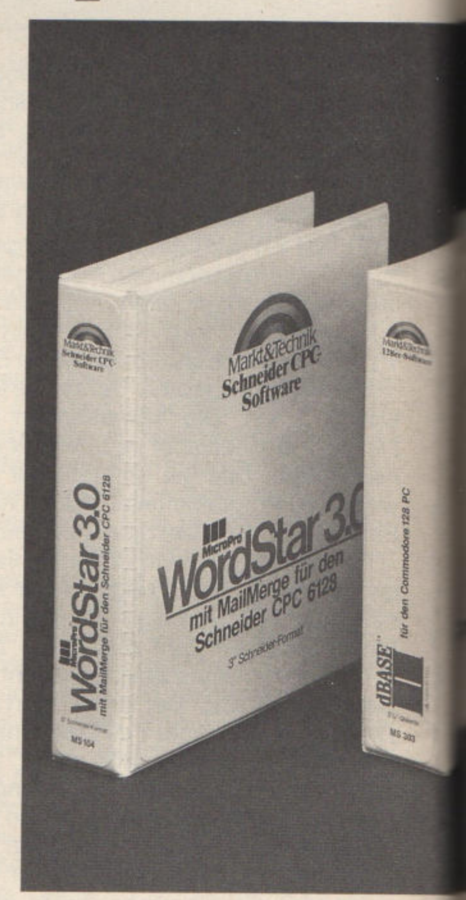
**MULTIPLAN, Version 1.06**
Wenn Sie die zeitraubende manuelle Verwaltung tabellarischer Aufstellungen mit Bleistift, Radiergummi und Rechenmaschine satt haben, dann ist MULTIPLAN, das System zur Bearbeitung »elektronischer Datenblätter«, genau das Richtige für Sie! Das benutzerfreundliche und leistungsfähige Tabellenkalkulationsprogramm kann bei allen Analyse- und Planungsberechnungen eingesetzt werden.

Sie erhalten jedes **WordStar-, dBASE II- und MULTIPLAN-Programm** für Ihren Schneider-Computer oder Commodore 128 PC fertig angepaßt (Bildschirmsteuerung). Jeweils Originalprodukte! Jedes Programmpaket enthält außerdem ein ausführliches Handbuch mit kompakter Befehlsübersicht.

| Version | Format | Bestell-Nummer | | |
|---|---|---|---|---|
| | | WordStar | dBase | Multiplan |
| Schneider CPC 464*/664* | 3" | MS 101 | MS 301 | MS 201 |
| Schneider CPC 464*/664* | 5¼" | MS 102 | MS 302 | MS 202 |
| Schneider CPC 6128 | 3" | MS 104 | MS 304 | MS 204 |
| Schneider Joyce | 3" | MS 105 | MS 305 | MS 205 |
| Commodore 128 | 5¼" | MS 103 | MS 303 | MS 203 |

*dBASE II und MULTIPLAN für die Schneider CPC 464/664 sind nur lauffähig mit einer Speichererweiterung auf mindestens 128 KByte und einer CP/M-Version für 62 KByte.

**Für Atari St:**
WordStar 3.0 (MS 106, DM 199,-*), dBase II (MS 306, DM 348,-*)




Diese Markt & Technik-Softwareprodukte erhalten Sie
in den Computer-Abteilungen der Kaufhäuser, bei Ihrem Computerhändle
im Buchhandel oder direkt beim Verlag gegen Vorauskasse

608201

# agbar!

**hnik**

# PLAN – für CP/M Computer













Jedes Buch kostet
**DM 49,–**
**(sFr. 45,10/öS 382,20)**
Erhältlich bei Ihrem Buchhändler.

**Und dazu die weiterführende Literatur:**
**WordStar für den Schneider CPC**
Best.-Nr. MT 779, ISBN 3-89090-180-8
WordStar für den Commodore 128 PC
Best.-Nr. MT 780, ISBN 3-89090-181-6
**dBASE II für den Commodore 128 PC**
Best.-Nr. MT 838, ISBN 3-89090-189-1
dBASE II für den Schneider CPC
Best.-Nr. MT 90188, ISBN 3-89090-188-3
**MULTIPLAN für den Schneider CPC**
Best.-Nr. MT 835, ISBN 3-89090-186-7
MULTIPLAN für Commodore 128 PC
Best.-Nr. MT 836, ISBN 3-89090-189-1

**Hardware-Anforderung für Schneider-Computer:**
Schneider CPC 464, CPC 664, CPC 6128, Joyce, beliebiger Drucker mit Centronics-Schnittstelle.
**Hardware-Anforderung für Commodore 128 PC:**
Commodore 128/128 D, Diskettenlaufwerk, 80-Zeichen-Monitor, Commodore-Drucker oder Drucker mit Centronics-Schnittstelle (ohne zwischengeschaltetes Interface).
**Übrigens gibt es WordStar, dBase und Multiplan auch für NDR-Computer.** Zu beziehen bei Graf Elektronik Systeme GmbH, Magnusstr. 13, 8960 Kempten.




Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2, 8013 Haar bei München

Bestellungen im Ausland bitte an untenstehende Adressen.
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. (042) 41 56 56
Österreich: Ueberreuter Media Handels- und Verlagsges. mbH., Alser Str. 24, A-1091 Wien, Tel. (0222) 4815 38-0

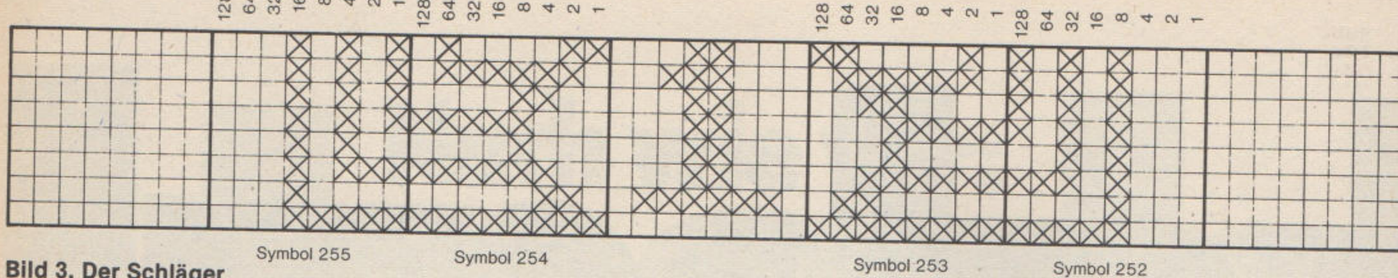



**Bild 3. Der Schläger**

(Listing 5) ist Dreh- und Angelpunkt des Programms. Mit EVERY wird eine Zeiteinheit definiert, nach der das Unterprogramm ab Zeile 7000 aufgerufen werden soll. Wir ziehen vom Wert 10 die Spielstufe (zu Anfang 1) ab und rufen mit diesem Wert die Ballroutine auf. Der hier angegebene Wert sagt, alle wieviel Fünfzigstel-Sekunden der Interrupt behandelt werden soll. Unabhängig von der Programmzeile wird die Ballroutine zu Anfang alle (10–1)-Fünfzigstel-Sekunden aufgerufen. Die »0« hinter dem Komma bezeichnet dem Computer die Uhr (eine von vier), die benutzt wird. Ab Zeile 7000 wird nun die Bewegung des Balles per Interrupt gesteuert. Zuerst wird die alte Position gelöscht, dann die neue Position berechnet (Zeile 7030 und 7040) und zum Schluß das Zeichen 231 als Ball gesetzt.

Am Rand unseres Bildschirms muß der Ball seine Richtung ändern, damit er diesen nicht verläßt. Dazu dienen die IF-Abfragen. Sie sagen nichts anderes, als »Wenn eine bestimmte Position erreicht ist, dann ändere die Laufrichtung«. Das erreicht man am einfachsten, indem man vor die entsprechende Variable ein Minuszeichen setzt. »verschiebungy« kennzeichnet dabei die Bewegung nach oben und unten – »verschiebungx« die nach rechts oder links. Damit der Aufprall auch zu hören ist, steht in der Zeile noch der Befehl »PRINT CHR$(7); (gibt einen Piepston aus)«. Mit einem »:« darf man jederzeit Befehle an schon geschriebene Zeilen anhängen. Unsere IF-Anweisung erweitert sich also um: »Wenn du an den Rand stößt, dann ändere die Richtung und gib einen Ton aus«.

Der LOCATE-Befehl setzt die Ausgabe-Position für den Schläger wieder auf den alten Wert. Ohne Zeile 7110 kann es passieren, daß der Schläger an der Ballposition erscheint. Der Interrupt muß nur unpassend »zuschlagen«. RETURN in Zeile 7120 sagt dem Computer, daß der Interrupt zu Ende ist.

So langsam bekommt unser Programm Gesicht. Allerdings wandert der Ball noch ohne Rückschlag durch die Mauer. Zeile 7070 aus Listing 6 fragt im Feld nach, ob an der aktuellen Ballposition ein Stein sitzt und verändert gege-

```
1000 REM Vorbereitung                      [61C2]
1040   MODE 1                              [7936]
2060 REM Rand                              [52E2]
2070   PRINT CHR$(150);                    [55F2]
2080   FOR i=2 TO 39                       [FBB0]
2090     PRINT CHR$(154);                  [C77E]
2100   NEXT i                              [D1D6]
2110   PRINT CHR$(156);                    [50F4]
2120   FOR i=2 TO 24                       [1C9A]
2130     LOCATE 1,i                        [4208]
2140     PRINT CHR$(149);                  [5F7E]
2150     PRINT TAB(40);CHR$(149);          [520E]
2160   NEXT i                              [ADE2]
2170 GOTO 2170                             [921A]
```

**Listing 1. Modus 1 ist unser Modus**

```
1020   DIM feld(40,25)                     [B400]
1030   INK 0,0:INK 1,26:INK 2,20:INK 3,1
    0                                      [B956]
2170 REM Bausteine                         [821C]
2180   FOR i=7 TO 13                       [E9AC]
2190     PEN (INT(i/3)-1)                  [5D4A]
2200     FOR j=2 TO 39                     [C926]
2210       LOCATE j,i                      [10F8]
2220       PRINT CHR$(233);                [D6F0]
2230       feld(j,i)=1                     [4800]
2240     NEXT j                            [2862]
2250   NEXT i                              [0AE2]
2260   PEN 1                               [1CBC]
```

**Listing 2. Der Bildschirm ist vierfarbig**

```
1060   ball=1                              [3A60]
1080   GOSUB 9000                          [9324]
2010 REM Aussehen des Schlaegers           [B440]
2020   SYMBOL 255,21,21,21,21,20,23,16,3
    1                                      [7A68]
2030   SYMBOL 254,67,126,12,248,8,252,6,
    255                                    [4A80]
2040   SYMBOL 253,194,126,48,31,16,63,96
    ,255                                   [EFF0]
2050   SYMBOL 252,168,168,168,168,40,232
    ,8,248                                 [75C6]
4010   LOCATE posx,25                      [6BB4]
4020   PRINT CHR$(32);CHR$(255);CHR$(254
    );CHR$(ball+48);CHR$(253);CHR$(252)
    ;CHR$(32);                             [AC28]
4070 GOTO 4070                             [BE1E]
9000 REM Startwerte                        [8844]
9050   posx=16                             [2D38]
9070 RETURN                                [CBA0]
```

**Listing 3. Bis auf den Ball ist alles zu sehen**

```
4000 REM Hauptprogramm                     [049E]
4030   posx=posx+(1+INKEY(1))-(1+INKEY(8
    ))                                     [3C6A]
4040   posx=posx-(posx=0)+(posx=35)        [A132]
4070 GOTO 4000                             [D310]
```

**Listing 4. Das Hauptprogramm ist kurz**

```
1050   spielstufe=1                        [7EB0]
3000 REM Interrupt initialisieren          [635A]
3010   EVERY 10-spielstufe,0 GOSUB 7000    [0908]
7000 REM Ballsteuerung ueber Interrupt     [37D0]
7010   LOCATE ballposx,ballposy            [3FEE]
7020   PRINT " "                           [1674]
7030   ballposx=ballposx+verschiebungx     [ECB2]
7040   ballposy=ballposy+verschiebungy     [26BA]
7050   LOCATE ballposx,ballposy            [3FF6]
7060   PRINT CHR$(231);                    [89FA]
7080   IF ballposx=2 OR ballposx=39 THEN
    verschiebungx=-verschiebungx:PRINT
    CHR$(7);                               [9D3E]
7090   IF ballposy=2 THEN verschiebungy=
    -verschiebungy:PRINT CHR$(7);          [F468]
7100   IF ballposy=24 THEN verschiebungy
    =-verschiebungy                        [AD6A]
7110   LOCATE posx-(posx=0)+(posx=35),25
                                           [97FC]
7120 RETURN                                [B694]
9010   ballposx=9                          [570A]
9020   ballposy=14                         [5F66]
9030   verschiebungx=1                     [2C2E]
9040   verschiebungy=1                     [9732]
```

**Listing 5. Crème de la crème – der Interrupt**

```
1070   stein=0                             [0770]
7070   IF feld(ballposx,ballposy)=1 THEN
    feld(ballposx,ballposy)=0:stein=st
    ein+1:verschiebungy=-verschiebungy:
    PRINT CHR$(7);                         [07A2]
```

**Listing 6. Die Mauer fällt**

```
7100   IF ballposy>=24 THEN GOSUB 8000     [C3EA]
8000 REM Untere Zeile                      [46F0]
8010   IF posx+1<=ballposx AND posx+5>=b
    allposx THEN verschiebungy=-verschi
    ebungy:PRINT CHR$(7); ELSE flag=1      [6BD4]
8030 RETURN                                [8796]
9060   flag=0                              [9B6C]
```

**Listing 7. Treffer – oder nicht?**

```
4050   IF flag=1 THEN GOTO 5000                    [A9DA]
5000 REM Neuer Ball                               [BF06]
5010   i=REMAIN (0)                               [4352]
5020   LOCATE ballposx,ballposy                   [E1EC]
5030   PRINT " ";                                 [36E8]
5040   LOCATE ballposx,25                         [51F2]
5050   PRINT CHR$(231);                           [63F4]
5060   LOCATE ballposx,25                         [45F6]
5070   PRINT " ";                                 [8EF0]
5080   LOCATE posx,25                             [59C4]
5090   PRINT SPACE$(7);                           [EB5C]
5100   GOSUB 9000                                 [701E]
5110   ball=ball+1                                [F9EC]
5120   IF ball=10 THEN CLS:PRINT "Ende !
   ":END                                          [F4B4]
5130   LOCATE posx,25                             [68BC]
5140   PRINT CHR$(32);CHR$(255);CHR$(254
   );CHR$(ball+48);CHR$(253);CHR$(252)
   ;CHR$(32);                                     [F930]
5150   IF INKEY$<>" " THEN GOTO 5150              [30DE]
5160 GOTO 3000                                    [7E10]
```

**Listing 8. Der Ball ist weg**

```
4060   IF stein=266 THEN GOTO 6000                [ADCA]
6000 REM Naechste Spielstufe                      [90B2]
6010   i=REMAIN (0)                               [5954]
6020   IF spielstufe=5 THEN MODE 2:PRINT
   "Ende !":END                                   [73D8]
6030   spielstufe=spielstufe+1                    [5094]
6040   stein=0                                    [7D74]
6050   GOSUB 9000                                 [0528]
6060   IF INKEY$<>" " THEN GOTO 6060              [E4E2]
6070 GOTO 2170                                    [7020]
```

**Listing 9. Geschafft – die nächste Spielstufe ist fällig**

```
1010   DEFINT a-z                                 [B008]
2000 REM Bildaufbau                               [00AA]
8020   IF (flag=0 AND (INT(RND*2)=0)) TH
   EN LOCATE ballposx,ballposy:PRINT "
   ";:ballposy=23                                 [C508]
```

**Listing 10. Feinheiten machen das Programm perfekt**

benenfalls die Flugrichtung. Gleichzeitig wird das Element des Feldes auf 0 gesetzt (der Stein ist ja weg), die Zahl der Steine um 1 hochgezählt und der Ton für den Aufprall ausgegeben. Die Variable »stein« wird in Zeile 1070 definiert. Mit ihrer Hilfe stellen wir später fest, ob die ganze Mauer abgeräumt ist.

Der Ball bewegt sich und der Schläger bewegt sich. Allerdings prallt der Ball auf dem Boden unabhängig vom Schläger zurück. Ändern wir 7100 in einen Unterprogrammaufruf zur Zeile 8000 (Listing 7), so können wir dort auswerten, ob der Schläger getroffen wurde oder nicht. Dazu definieren wir in Zeile 9060 ein Flag, das immer dann den Wert 1 hat, wenn der Ball nicht auf den Schläger trifft. Zuerst ist es deshalb natürlich 0. Wenn die Schlägerposition »posx+1« kleiner als die Ballposition »ballposx« oder »posx+5« größer als »ballposx« ist, dann trifft der Ball auf den Schläger, die Richtung wird verändert und der Ton ausgegeben. Andernfalls trifft die Anweisung nach dem Befehl ELSE zu: »flag« wird auf 1 gesetzt. Daß der Wert »posx+1« ausschlaggebend ist, liegt daran, daß rechts und links neben den Schläger immer ein Leerzeichen zum Übermalen der alten Position gezeichnet wird. Damit übermalt der Computer die alte Position des Schlägers.

Das Flag, das den »Nichttreffer« kennzeichnet, wird in dem Programmteil ab Zeile 5000 (Listing 8) ausgewertet. Fügen Sie Zeile 4050 in den Hauptteil ein, und schon ruft der Interpreter immer, wenn das Flag gesetzt ist, das Unterprogramm auf. Zuerst wird der Interrupt gelöscht (der Ball darf ja nicht »dazwischenfunken«). Der Befehl in Zeile 5010 ist eine Hilfskonstruktion. Immer dann, wenn mit REMAIN der Restwert der Interrupt-Uhr (das ist die Zeit, die bis zum nächsten Interruptaufruf noch vergeht) ausgegeben wird, wird diese auch gelöscht.

Der Ball fliegt nun nicht mehr. Als nächstes wird er in die Zeile 25 geschrieben (er fällt ja auf die Erde). Natürlich wurde er zuvor in Zeile 24 gelöscht. Auch der Schläger ist im Moment überflüssig. Mit »PRINT SPACE$(7);« werden an der Stelle des Schlägers 7 Leerzeichen ausgegeben. Die Werte aus der Routine in Zeile 9000 müssen wieder restauriert werden (»GOSUB 9000«) bevor der Computer den Spieler fragen kann, ob er mit dem nächsten Ball fortfahren soll. Vorher muß der Wert »ball« um 1 heraufgesetzt und, falls dieser dann 10 erreicht, das Spiel abgebrochen werden.

Zur nächsten Runde malt das Programm den Schläger auf die Ausgangsposition, bevor es in der Zeile 5150 darauf wartet, daß Sie die Leertaste drücken. Diesmal benutzten wir zur Tastaturabfrage den Befehle INKEY$. Dieser liest nämlich das Zeichen, das auf einer Taste liegt, direkt ein. »INKEY$="A"« ist also nur aktiv, wenn das große A gedrückt ist. Solange INKEY$ kein Leerzeichen findet, bleibt das Programm in Zeile 5150 hängen. Ansonsten wird mit Zeile 3000 weitergemacht.

Was passiert nun, wenn die Mauer abgeräumt ist? Die Spielstufe muß erhöht und die Mauer wieder aufgebaut werden. Die Mauer hat genau 266 (=7x38) Steine; in der Variablen »stein« zählt der Computer die Treffer. Fügen Sie nun Zeile 4060 (aus Listing 9) in das Hauptprogramm ein, dann behandelt die Routine ab 6000 das Ende einer Spielstufe. Das Unterprogramm ähnelt etwas dem ab Zeile 5000. Zuerst wird der Interrupt gelöscht, dann die Spielstufe auf ihren Maximalwert getestet (größer als 5 darf sie nicht werden, da sonst der Interrupt so schnell zuschlägt, daß Sie keine Chance haben, den Ball zu treffen) und gegebenenfalls das Spiel beendet. Falls das Programm in der nächsten Spielstufe weiterarbeitet, setzen wir »spielstufe« hoch und die Ausgangswerte mit »GOSUB 9000« ein. In Zeile 6050 wartet der Computer wieder auf den Druck der Leertaste.

| gedrückte Taste | Rückgabewert |
|---|---|
| Taste nicht gedrückt | -1 |
| Taste allein | 0 |
| Taste mit <SHIFT> | 32 |
| Taste mit <CTRL> | 128 |
| Taste mit <SHIFT> und <CTRL> | 160 |

**Tabelle 1. Mit diesen Informationen werten Sie die Tastatur aus**

Mit »GOTO 2170« kehrt das Programm an den Anfang zurück – und zwar dorthin, wo der Computer die Mauer »baut«.

Listing 10 fügt nur noch letzte Verbesserungen ein. Wenn Sie die Variablen mit DEFINT auf Integerwerte festlegen, beschleunigt dies die Bearbeitungsgeschwindigkeit enorm. Das interne Format von Ganzzahlen ist nämlich für den Interpreter leichter – und damit auch schneller – zu managen als das der reellen Zahlen. Sicher ist Ihnen auch aufgefallen, daß unsere Mauer nur rasterförmig eingerissen wird. Die Symmetrie des Programms ändern wir, indem wir per Zufall gesteuert beim Treffen auf den Schläger den Ball eine Bildschirmzeile senkrecht nach oben steuern. Dazu dienen die Anweisungen in 8020.

Falls der Ball nicht auf den Boden fällt, gilt (»flag=0«) und »INT(RND*2)=0«. In diesem Fall löscht der Computer die alte Ballposition und setzt »ballposy« auf 23. Mit der Anweisung rufen Sie eine Zahl zwischen einschließlich 0 und ausschließlich 1 auf. Welche, das hängt von der Zeit ab, die vergangen ist, seitdem der Computer eingeschaltet wurde. Und dieser Wert ist wirklich nahezu unberechenbar und damit zufällig. Multiplizieren Sie den Wert mit 2, so bekommen Sie immer eine Ziffer zwischen 0 und 2. Der Vorkommawert davon ist entweder 0 oder 1. Der Wahrscheinlichkeit nach sind beide Möglichkeiten gleich häufig. Die Chance, daß der Ball um eine Zeile verschoben wird, beträgt also 50 Prozent. (hg)

# Der CPC als Tor zur Welt

**Alles Wissen dieser Welt auf dem Terminal Ihres Schneider-Computers: Ein Akustikkoppler, eine serielle Schnittstelle und unser Terminalprogramm ist alles, was Sie dazu brauchen.**

Wie ein unsichtbares Spinnennetz haben sich seit Mitte der siebziger Jahre die elektronischen Datennetze über die Erde gespannt. Fertigungsdaten für Industrieroboter werden darüber genauso ausgetauscht wie aktuelle Wetterinformationen, Aktienkurse, politische Sensationen, Smalltalk, Halbwertzeiten beim radioaktiven Zerfall und Kochrezepte. Für einen Anwender, der mit seinem Computer schnell große Mengen an Daten verarbeiten will und auch den Zugang auf externe Informationen braucht, ist die Datenfernübertragung die logische Erweiterung von Textverarbeitung, Datenbank und Spreadsheet.

Eine Datenbank funktioniert ähnlich der Adreßverwaltung auf dem Heimcomputer: Der Computer kann eine große Menge von Datensätzen in relativ kurzer Zeit durchsuchen, sortieren und nach bestimmten Kriterien geordnet wieder ausgeben.

Der wesentliche Unterschied zwischen Datenbank und Heimcomputer liegt einmal in der Größe der Datei (mehrere zehn- oder gar hunderttausend Einträge sind keine Seltenheit) und zum anderen darin, daß die Arbeit mit einer Datenbank über die Telefonleitung (»online«) vor sich geht.

Rund fünftausend Datenbanken und mehrere zehntausend Mailboxen können Sie mit Ihrem CPC weltweit erreichen. Datenbanken gibt es zu fast jedem denkbaren Themenkomplex: Juristische, betriebswirtschaftliche, medizinische, chemische, sozialwissenschaftliche und technische Datensammlungen mit oft mehreren hundert Megabyte können Sie zu Hause am Schreibtisch durchstöbern. Sofern man das nötige Kleingeld hat. Etwa tausend Mark je Stunde Benutzungszeit kosten beispielsweise amerikanische Datenbanken, die sich auf aktuelle Bodenschatzfunde spezialisiert haben.

Für den Hobbyanwender erschwinglich sind Mailboxen, elektronische Briefkästen, die ausgesprochene Enthusiasten betreiben und die in der Regel nichts kosten. Allein in der BRD gibt es mittlerweile rund vierhundert solche nichtkommerzielle Systeme. Das technische Prinzip ist einfach: Ein Computer hängt an einem öffentlich zugänglichen Netz (entweder am Telefonnetz oder an DATEX-P). Jeder eingetragene Benutzer kann ihn anrufen und hat (unterschiedlichen) Zugriff auf verschiedene dort gespeicherte Files. Er kann auch Daten hineinschreiben, die andere Benutzer wieder lesen können. Je nachdem, ob diese geschriebenen Dateien allen zugänglich sind oder nur einem bestimmten Benutzer, spricht man im übertragenen Sinne von elektronischen »Schwarzen Brettern« oder von »privaten Postfächern«. In der Regel sind diese verschiedenen Funktionen aus einem Menü auszuwählen, das die Mailbox nach dem Einloggen (der erfolgreichen Eingabe von Benutzername und Paßwort) anbietet.

Wie sinnvoll und arbeitssparend eine Mailbox auch für den Privatmann und die Privatfrau sein kann, zeigt folgendes Beispiel: Michael braucht dringend die technische Beschreibung eines elektronischen Bauteils für eine Seminararbeit. Diese Beschreibung will ihm Peter liefern. Peter ist nur unglücklicherweise gerade zu Besuch bei seiner Großmutter und die Sache ist dringend, so daß ein Brief zu spät ankäme. Hat Peter bei seiner Großmutter »zufälligerweise« seinen Computer und einen Akustikkoppler dabei, dann ist alles kein Problem: Er ruft Michael an und schickt ihm den Text, den er vorher geschrieben hat, direkt in seinen Computer. Das hat im Gegensatz zum Brief den Vorteil, daß Michael den Text nicht mehr eintippen braucht, sondern ihn direkt am Bildschirm mit Hilfe eines Textverarbeitungsprogramms editieren kann.

## Der Brief übers Telefon

Was aber tun, wenn aus irgendeinem Grund keine direkte Verbindung zwischen Michael und Peter möglich ist? Wenn Michael beispielsweise durch seine Arbeitszeit nur sehr schlecht zu erreichen ist? Dann benützt Peter eine Mailbox, in der beide eingetragene Benutzer sind. Er ruft an, gibt seinen Usernamen und sein Paßwort ein und wählt den Menüpunkt »Private Post«. Auf die Frage des Mailboxcomputers, an wen die Nachricht gehen soll, gibt Peter den Systemnamen von Michael ein, und hat nun die Möglichkeit, den Bericht, den Michael so dringend braucht, in dessen persönliches Postfach zu laden. Abholen kann Michael sich den Artikel wann er will, ob um vier Uhr früh oder neun Uhr abends (oder zu irgendeiner anderen nachtschlafenden Zeit). Texte versenden via Mailbox geht nicht nur wesentlich schneller als Eilbrief, Kurier oder Telex, es ist außerdem ja möglich, die Texte sofort weiterzuverarbeiten. Doch der wichtigste Vorteil ist der zeitunabhängige Versand. Sowohl Absender als auch Empfänger können den Nachrichtenverkehr dann aufnehmen, wann es ihnen in den Kram paßt – in Sekunden- oder Wochenabstand voneinander, ganz wie es beliebt.

Parallel dazu, wie die lokalen Netzwerke (LAN) in den Betrieben an Boden gewinnen, steigt auch die Bedeutung der Datenfernübertragung. Die niedrigen Kosten für die nötige Hardware und die unübersehbare Flut an preiswerter bis kostenloser DFÜ-Software bereiten den Weg zu einer Datenfernübertragung als Massenkommunikationsmittel. Die Hacker machen es vor: Neben kommerziellen Aspekten ist es auch ein potentielles Mittel zur Völkerverständigung. Wenn man sich für dreiundzwanzig Pfennig anschauen kann, was in den USA, in England oder in Nicaragua und der UdSSR gedacht wird, und mit den Menschen dort über die Netze reden und Nachrichten austauschen kann, wird dieses Verständnis anderer Völker Kriege unwahrscheinlicher machen.

Wie bekannt, kann man den Schneider nur schwer dazu bewegen die Daten seriell, das heißt Bit für Bit, zu übertragen. Da der Schneider nur TTL-Pegel ( 0 Volt und 5 Volt) von sich gibt, wird man sich schwertun, das Modem (oder den Akustikkoppler) davon zu überzeugen, die gesandten Informationen zu verstehen. Beide verlangen nämlich die Daten nach RS232C-Norm (+/–12 Volt).

Eine serielle Schnittstelle, wie sie beispielsweise Vortex anbietet, wandelt die unterschiedlichen Pegel um, genauso wie Akustikkoppler und Modem die Signale des CPC für das Telefon aufbereiten.

Stecken Sie die Schnittstelle in den Erweiterungsanschluß Ihres Schneiders, danach das Verbindungskabel in die Schnittstelle und in das Modem.

Wie alle kommerzielle Software sind auch Terminalprogramme relativ teuer. Für alle die, die nicht auf die kostenlosen Public-Domain-Programme unter CP/M zugreifen können, stellen wir hier ein Terminalprogramm vor, das sich auch nicht vor kommerziellen Produkten zu verstecken braucht.

Zur Zeit gibt es rund ein halbes Dutzend serielle Schnittstellen für den Schneider CPC. Von den selbstgestrickten ganz zu schweigen. Daher ist unser Programm so geschrieben, daß

man es sehr leicht anpassen kann. Je nach Schnittstelle reicht es, lediglich einige Zeilen auszutauschen.

**Folgende Programmteile müssen Sie abtippen:**

**Vortex:**
- Basicprogramm »Main« (Listing 1)
- Basicprogramm »vortex« (Listing 2)
- Maschinenprogramm »mcain« (Listing 5)
- Maschinenprogramm »mc0« (Listing 6)

**Schneider:**
- Basicprogramm »Main« (Listing 1)
- Basicprogramm »schneider« (Listing 3)
- Maschinenprogramm »mcain« (Listing 5)
- Maschinenprogramm »mc2« (Listing 7)

**eigene:**
- Basicprogramm »Main« (Listing 1)
- Basicprogramm »eigene« (Listing 4)
- Maschinenprogramm »mcain« (Listing 5)
- Maschinenprogramm »mc1« (Listing 8)

Die beiden Basicprogramme und die beiden DATA-Lader müssen jeweils als ein Programm eingegeben werden.

**Mailboxen mit Informationen für Schneider-Computer**

Eigene Mailboxbretter für den CPC mit Tips und Tricks, Informationen und Anfragen haben beispielsweise folgende Mailboxen:

| | |
|---|---|
| Toelleturm-Box | Tel.: 0202/559350 |
| Vollrath-Box | Tel.: 0209/271666 |
| Datenmühle | Tel.: 030/3219768 |
| ACM-Box | Tel.: 089/8120338 |
| Hitech | Tel.: 089/392289 |
| Schweiz: | |
| Hobby-Box | Tel.: 0041/1741/3314 |

Wenn Sie das Programm richtig abgetippt haben, gehen Sie beim Aufbau einer Verbindung folgendermaßen vor:

Sie wählen beispielsweise eine der obenstehenden Nummern. Den Mailboxcomputer am anderen Ende der Leitung erkennen Sie an dem charakteristischen hohen Pfeifton aus dem Telefonhörer. Nun legen Sie den Telefonhörer in den Koppler. Danach drücken Sie <D> und ein paarmal <ENTER>, damit der andere Computer weiß, daß jemand in der Leitung ist. Darauf meldet dieser sich mit seiner Einschaltmeldung. Wenn er sich nicht rührt, so überprüfen Sie, ob Ihr Koppler einen Ton von sich gibt. Falls nicht, überprüfen Sie die Stromversorgung und Verbindung zum Koppler. Falls alles funktioniert und der angerufene Computer sich trotzdem nicht meldet, kann es sein, daß das Telefonsignal zu schwach ist (noch mal anrufen) oder der Telefonhörer verkehrtherum im Koppler liegt.

Das Programm ist kommandogesteuert. Mit **»h«** gelangt man beispielsweise in den Hilfsmodus, in dem alle Kommandos erklärt werden.

Bei Modems mit Selbstwahl gibt man die Nummer einfach über die Tastatur ein. Nach dem Drücken der ENTER-Taste schickt der CPC die entsprechenden Impulse an das Modem, und dieses stellt die Verbindung her. Anschluß besetzt? Einfach noch mal <ENTER>, und der Computer legt auf und wählt von neuem. Wenn Sie Telefonnummern oft brauchen, dann können Sie diese in Zeile 490 eingeben und brauchen dann statt der ganzen Nummer nur die gespeicherte Kurzwahl einzutasten.

**Folgende Befehle versteht unser Terminalprogramm:**

**»d«** (Dialogbetrieb) Geht in den Dialogmodus und teilt dem anderen Computer mit, daß er mit dem Senden anfangen kann. (Sendet »XON«.) XON wird gesendet, damit der andere Computer nach einem etwaigen XOFF zu senden beginnt.

**»TAB«** (Unterbrechung) Rückkehr in die Befehlsebene und Senden von »XOFF«, so daß der Mailboxcomputer aufhört zu senden.

**»s«** (speichern) Alle Ein- und Ausgaben der beiden beteiligten Computer werden in einen Puffer gespeichert.

**»dl«** (Dialogbetrieb mit Download) Es wird ein Protokoll auf Diskette mitgespeichert.

**»l«** (laden) Daten von Diskette in den Puffer laden. Falls schon Daten im Puffer waren, werden die neuen hinten angehängt.

**»c«** gibt das Inhaltsverzeichnis der Diskette aus.

**»td«** sendet einen Text direkt von der Diskette an den anderen Computer.

**»type«** Inhalt eines Files auf der Diskette auf den Bildschirm ausgeben.

**»w«** (Warmstart) Bildschirm löschen und Farben zurücksetzen.

**»r«** (read) Pufferinhalt auf den Bildschirm ausgeben. Auf Tastendruck wird die Ausgabe gestoppt und wieder fortgesetzt.

**»n«** (Nummer) Mit »n1« wird die erste Nummer der Telefonliste gewählt (nur mit Modem).

**»p«** (Parameter ändern) Damit sich die zwei Computer verstehen, wird ein sogenanntes Übertragungsprotokoll vereinbart, das die Anzahl der Datenbits, die Parität und die Stopbits festlegt. Hier sollte man herumprobieren, wenn man statt einer vernünftigen Verbindung nur sinnlosen Schrott auf den Schirm bekommt. Im Menüpunkt »p« kann man mit **»?«** andere Übertragungsprotokolle einstellen und mit **»?p«** den Schnittstellenbaustein direkt programmieren. Es können zum Beispiel unterschiedliche Sende- und Empfangsbaudzahlen eingestellt werden, so daß der CPC mit 1200/75 Baud auch BTX-fähig wird. Gibt man nur <Return> ein, so bleiben die eingestellten Parameter unverändert.

**»voll«** und **»halb«** stellen auf Voll- und Halbduplex um.

**»e«** beendet das Programm.

(Klaus Bell/jg)







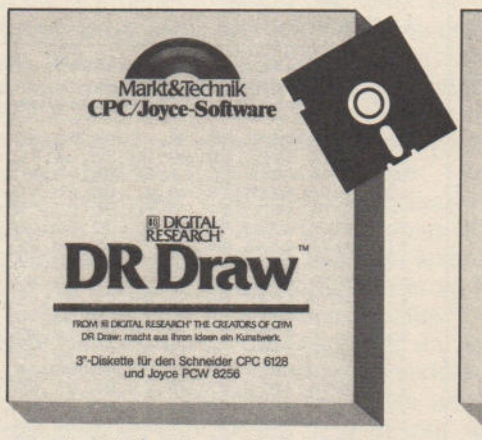


```
10 'DFUE Programm von Bell Klaus 08102/1
   067
20 '*********  Initialisierung          [75F6]
21 MEMORY &3000:OPENOUT "dummy":MEMORY H  [757A]
   IMEM-1:CLOSEOUT:DEFINT h             [40FC]
23 WINDOW#1,10,80,1,25                  [0484]
24 ON BREAK GOSUB 620                   [A700]
25 ON ERROR GOTO 1000                   [C510]
26 FOR a=1 TO 3:READ par$(a),stbit$(a):N
   EXT                                  [C148]
27 DATA keine,1,ungerade,1 1/2,gerade,2 [6948]
74 POKE base+&10F,&24:adr=base+&6:by=&30
   00:GOSUB 1700:POKE base+&62,0        [0116]
75 PRINT"<20>DFUE-Programm<20>(h-Help)  [9680]
76 PRINT"<19>---------------            [E47E]
77 PRINT                                [6A36]
80 GOTO 500                             [37EC]
81 '-------------------------------------
   ---                                  [6616]
82 '           HAUPTTEIL                [A5C2]
83 '-------------------------------------
   ---                                  [0C1A]
90 INPUT "Befehl: ",a$:a$=LOWER$(a$)    [699A]
100 IF a$="" THEN a$=b$                 [953A]
105 b$=a$                               [01FC]
110 IF a$="e" THEN MODE 2:END           [5D86]
115 IF a$="info"THEN GOTO 700           [26AE]
120 IF ASC(a$)=110 THEN a=VAL(RIGHT$(a$,
    LEN(a$)-1)):a$=tel$(a)              [22DA]
125 IF a$="io" THEN PRINT"Ein/Ausgabedat
    ei bestimmen":GOTO 1200 .           [5C92]
130 IF a$="era" THEN INPUT"filename ";na
    m$:|ERA,@nam$:GOTO 90               [506C]
135 IF ABS(ASC(a$)-53)<6 THEN 205       [FD92]
140 IF a$="c"THEN CAT:GOTO 90           [7EE0]
144 IF a$="puffer" THEN INPUT "Wieviele
    Bytes ";by:by=by+pa:adr=base+4:GOSUB
     1700:GOTO 90                       [7ABE]
145 IF a$="w" THEN CALL &BB4E:MODE 2:GOT
    O 90                                [45C2]
150 IF a$="s" THEN INPUT"filename ";nam$
    :OPENOUT nam$:CALL base+&21:CLOSEOUT
    :POKE base+&4,0:POKE base+&5,0:GOTO
    90                                  [7DEA]
155 IF a$="p" THEN INPUT "Parameter eing
    eben:";par$:GOTO 1300               [3EE8]
156 IF a$="r" THEN INPUT "Puffer lesen.
    Ab Byte ";by$:GOTO 1500             [2B7A]
157 IF a$="voll" THEN du$=" Fullduplex":
    POKE base+&10F,&24:GOTO 90          [6162]
158 IF a$="halb" THEN du$=" Halfduplex":
    POKE base+&10F,&21:GOTO 90          [33E2]
159 IF a$="filter" THEN INPUT "Mit welch
    em Byte soll gefiltert werden ";by:P
    OKE base+&59,by:GOTO 90             [F682]
160 IF a$="zeig" THEN INPUT"filename ";n
    am$:PRINT"Searching for ";nam$:OPENI
    N nam$:GOTO 400                     [C68A]
161 IF a$="lfon" THEN lf=10:POKE base+&6
    2,10:GOTO 90                        [B0C2]
162 IF a$="lfoff" THEN lf=0:POKE base+&6
    2,0:GOTO 90                         [6BBC]
165 IF a$="l" THEN INPUT"filename ";nam$
    :PRINT"Searching for ";nam$:POKE bas
    e+&C0,&CF:OPENIN nam$:CALL base+&D6:
    CLOSEIN:POKE base+&C0,&C6:GOTO 90   [DEEA]
170 IF a$="tel"THEN GOTO 510            [13E0]
175 IF a$="i" THEN GOTO 900             [1778]
180 IF a$="h" THEN GOTO 520             [A16A]
189 IF a$="td" THEN INPUT"filename ";nam
    $:PRINT"Searching for ";nam$:GOTO 87
    0                                   [FD02]
190 IF a$="d" THEN POKE base+&C0,&C6:GOS
    UB 1400:PRINT:GOTO 90               [7B3A]
195 IF a$="dl" THEN POKE base+&C0,&CF:IN
    PUT "Filename";nam$:OPENOUT nam$:adr
    =base+4:by=pa:GOSUB 1700:GOSUB 1400:
    CALL base+&21:CLOSEOUT:PRINT:GOTO 90
                                        [1D8C]
200 GOTO 90                             [DDE8]
205 '**************   Auflegen, Abheben
                                        [0A98]
206 OUT &F6FF,16:FOR l=1 TO 300:NEXT    [7636]
207 OUT &F6FF,0:FOR l=1 TO 1200:NEXT    [372A]
209 '*************     Waehlen          [84E4]
210 PRINT CHR$(13);CHR$(11);"Nummer: ";a
    $;"<27>"                            [49E2]
220 FOR z=1 TO LEN (a$):c=ASC(MID$(a$,z,
    1))                                 [E868]
230 IF c<48 OR c>57 THEN GOTO 260       [5F7A]
240 c=c-48:IF c=0 THEN c=10             [E0F6]
250 GOSUB 300                           [53D4]
260 NEXT                                [64EE]
270 GOTO 90                             [A4F6]
280 GOSUB 1400:GOTO 90                  [4C36]
300 '*************     Nummer Waehlen   [1ABC]
305 FOR a=1 TO c                        [C6A8]
310 FOR l=1 TO 49                       [0CCA]
320 NEXT                                [73E8]
330          OUT &F6FF,16:'  Port c von
     8255 Bit 4=high  (Relai oeffnen)   [D9DC]
340 FOR l=1 TO 40                       [F5BE]
350 NEXT                                [6BEE]
360          OUT &F6FF,0:'  Bit 4=low:
      Relai schliessen                  [32BA]
370 IF INKEY$<>"" THEN OUT &F6FF,0:GOTO
    90:'Relai schliessen und nach Befehl
    sebene                              [EBE0]
380 NEXT                                [7FF4]
390 FOR l=1 TO 400:NEXT                 [551A]
395 RETURN                              [9942]
400 '*****************    Type          [8262]
401 WHILE NOT EOF                       [5572]
420 LINE INPUT #9,t$:PRINT t$           [309A]
430 a$=INKEY$:IF a$="" THEN GOTO 430 ELS
    E IF a$=" " THEN CLOSEIN:GOTO 90    [4142]
435 IF a$="w" THEN PEN 1:PAPER 0:BORDER
    0:INK 1,24:INK 0,0                  [4538]
440 WEND                                [2FCC]
450 PRINT                               [648C]
460 CLOSEIN                             [1D8E]
470 GOTO 90                             [B8FA]
480 '*******************  Telefonliste
    Format: Name,Nummer                 [3DA6]
481 DATA Hitech,089/392289              [E412]
490 DATA Datex P,089/228730             [C41E]
500 da=2:DIM nam$(da):DIM tel$(da):RESTO
    RE 480:FOR a=1 TO da:READ nam$(a),te
    l$(a):NEXT:GOTO 90                  [14B8]
510 FOR a=1 TO da:PRINT USING"##";a;:PRI
    NT") ";nam$(a),tel$(a):NEXT:GOTO 90 [CEBE]
520 '*****************    Help - Funktion
                                        [CAAC]
521 CLS:CLS #1:WINDOW SWAP 1,0:         [47FC]
525 PRINT"<7>FUNKTIONEN                 [0698]
530 PRINT:PRINT"Enter<6>Wahlwiederholung
                                        [E90A]
534 PRINT"d<7>Dialog                    [D8FE]
536 PRINT"dl<6>Dialog mit Down Load     [851E]
540 PRINT"s<13>Daten saven              [DB48]
545 PRINT"td<12>Text abschicken von Disc
                                        [2556]
550 PRINT"l<13>Daten laden              [DC0A]
551 PRINT"halb<10>Halbduplex            [5294]
552 PRINT"voll<10>Vollduplex            [1E2E]
553 PRINT"lfon/lfoff<4>Linefeed nach CR<
    2>ja oder nein                      [9702]
555 PRINT"r<13>Puffer lesen             [C93E]
560 PRINT"tel<5>Telefon: -Verzeichnis   [F65A]
565 PRINT":<16>-abschalten (=auflegen)  [1CC0]
570 PRINT"n<16>-Id fuer Nummer eingeben [B4EC]
576 PRINT"io<12>Ein/Ausgabeeinheit defin
    ieren                               [46D6]
580 PRINT"info<10>Informationen         [BF62]
585 PRINT"puffer<8>Pufferzeiger verbiege
    n                                   [117C]
589 PRINT"h<13>HELP                     [794C]
590 PRINT"c<13>CAT                      [BC90]
595 PRINT"p<13>Parameter aendern        [4F40]
596 PRINT"w<13>Warmstart                [2A9E]
600 PRINT"e<13>Ende                     [2DCC]
610 WINDOW SWAP 1,0:LOCATE 1,25:GOTO 90 [D6D2]
620 OUT &F6FF,0:GOTO 90                 [86AC]
700 '**********************   INFO      [0774]
701 frei=PEEK(base+&4)+256*PEEK(base+&5)
    -pa                                 [4E70]
702 PRINT:PRINT"Informationen:":PRINT   [9656]
710 PRINT"Bytes im Puffer: ";frei       [4192]
716 PRINT"Bytes frei:"pg-frei           [4F80]
717 PRINT"Himem: ";HIMEM                [C40A]
718 PRINT"Puffergroesse ";pg            [9246]
719 PRINT"Parameter:"                   [E2DA]
720 PRINT bd;" baud Senderate           [7900]
721 PRINT bd2;" baud Empfangsrate       [14EA]
723 PRINT dat;" Datenbits               [E87A]
724 PRINT " ";par$(par+1);" Paritaet    [A52E]
725 PRINT " ";stbit$(stbit+1);" Stopbit(
    s)                                  [F09A]
726 PRINT DU$                           [4A52]
730 GOTO 90                             [E0F8]
870 OPENIN nam$:POKE base+&17,0:CALL bas
    e+&8:POKE base+&17,&F0:CLOSEIN:GOTO
    90                                  [8886]
1000 PRINT"GFehlerG",ERR,ERL:RESUME 90  [7C9C]
1200 '*************** Ein / Ausgabe be
     stimmen                            [E736]
1201 PRINT:PRINT"1-Drive A              [9626]
1210 PRINT"2-Drive B                    [AE9C]
1220 PRINT"3-Tape speedwrite 1          [6456]
1230 PRINT"4-Tape speedwrite 2          [475C]
1240 INPUT a:IF a=1 THEN |DISC:|A ELSE I
     F a=2 THEN |DISC:|B ELSE IF a=3 THE
     N |TAPE:speedwrite 1 ELSE IF a=4 TH
     EN |TAPE:POKE &B8D1,2:POKE &B8D2,&1
     7 ELSE PRINT"Nichts veraendert     [231A]
1250 GOTO 90                            [2754]
1300 '**************   Parameter eingeb
     en                                 [44BE]
1301 IF par$="8n1" THEN dat=8:par=0:stbi
     t=0:GOTO 1380                      [540C]
1310 IF par$="8n2" THEN dat=8:par=0:stbi
     t=2:GOTO 1380                      [1B12]
1320 IF par$="7n1" THEN dat=7:par=0:stbi
```

**Listing 1. DFÜ mit einem eigenen Programm**




# Professionelle Grafikprogramme für Schneider CPC 6128 + Joyce



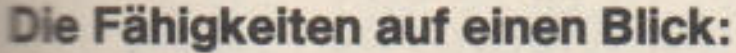






**DR Draw: Macht aus Ihren Ideen ein Kunstwerk.**

Verwenden Sie DR Draw, um Organisations-Diagramme, Flußdiagramme, Logos, technische Zeichnungen, Schaubilder, Platinenentwürfe und jede nur erdenkliche Art von Linien- und Formgrafiken zu entwerfen. Jeder Bestandteil Ihrer Zeichnung kann auf vielfältige Weise durch Farben und Schraffuren hervorgehoben werden.



**Die Fähigkeiten auf einen Blick:**

- Erstellung beliebiger Zeichnungen
- vorprogrammierte Figuren wie Kreise, Quader, Rechtecke, Kreisbögen, Polygone und Linien
- freie Wahl der Gestaltungselemente wie Farben, Muster und Schriftarten
- Vergrößerungen und Ausschnittdarstellungen
- Teile einer Zeichnung können kopiert, verschoben oder gelöscht werden
- Grafiken können gespeichert, geplottet oder gedruckt werden
- einfache Bedienung durch Menüauswahl

**Hardwarevoraussetzungen:**

DR Draw läuft auf jedem Schneider CPC 6128 oder Joyce PCW 8256 mit einem oder zwei Diskettenlaufwerken. Die Grafiken können auf jedem Drucker oder Plotter ausgegeben werden, für den ein GSX-Treiber verfügbar ist. Dazu zählen Schneider-, Epson- und Shinwa-Drucker sowie der Plotter HP 7470A.

Diese Markt & Technik-Software-Produkte erhalten Sie in den Computer-Abteilungen der Warenhäuser, bei Ihrem Computerfachhändler, im Buchhandel oder direkt beim Verlag gegen Vorauskasse.

**DR Graph: Präsentationsgrafiken mit professionellem Niveau.**

DR Graph ist ein interaktives Softwarepaket, mit dem Sie Ihren Mikrocomputer zur Erstellung von Geschäftsgrafiken und Text-Charts verwenden können.

**Die Fähigkeiten auf einen Blick:**

- Linien-Grafiken, Histogramme, Torten-Grafiken, Stufen-Grafiken, Strich-Histogramme, Punkte-Grafiken und Text-Grafiken
- freie Wahl der Gestaltungselemente wie Beschriftungen, Titelzeilen, Legenden, Farben, Schriftarten und Ränder
- frei wählbare Skalierung
- variable Linien- und Balkenbreite
- Schnittstelle zu anderen Programmen
- beliebig positionierbare Anmerkungen
- Grafiken können gespeichert, geplottet oder gedruckt werden
- einfache Bedienung durch Menüauswahl

**Hardwarevoraussetzungen:**

DR Graph läuft auf jedem Schneider CPC 6128 oder Joyce PCW 8256 mit einem oder zwei Diskettenlaufwerken. Die Grafiken können auf jedem Drucker oder Plotter ausgegeben werden, für den ein GSX-Treiber verfügbar ist. Dazu zählen Schneider-, Epson- und Shinwa-Drucker sowie der Plotter HP 7470A.

| | Version | Best.-Nr. | Format | Preis DM | sFr | öS |
|---|---|---|---|---|---|---|
| DR Draw | CPC 6128/Joyce | MS 613 | 3" | 199,-* | 178,- | 1990,-* |
| DR Graph | CPC 6128/Joyce | MS 614 | 3" | 199,-* | 178,- | 1990,-* |

* inkl. MwSt. Unverbindliche Preisempfehlung




612235

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Tel. (0222) 677526 · Ueberreuter Media Verlagsges. mbH, Alser Straße 24, A-1091 Wien, Tel. (0222) 481538-0.

```
        t=0:GOTO 1380                          [E90A]
 1322 IF par$="7n2" THEN dat=7:par=0:stbi
        t=2:GOTO 1380                          [F114]
 1324 IF par$="?" THEN GOTO 1600               [E19E]
 1326 IF par$<>"?e" THEN PRINT "Nichts ge
        andert":GOTO 90                        [F226]
 1330 PRINT"Parameter selber eingeben!<2>
        Zurueck mit ENTER"                     [714A]
 1334 INPUT"Welchen Offset";a$:IF a$="" T
        HEN GOTO 90 ELSE |BASE,2,@h:adr=h+V
        AL(a$)                                 [3330]
 1336 INPUT"Daten";a$                          [9BDA]
 1340 IF a$="" THEN GOTO 90                    [409C]
 1345 by=VAL(a$):IF d>255 THEN GOSUB 1700
        :GOTO 1334                             [3D34]
 1350 POKE adr,by:GOTO 1334                    [72A8]
 1400 CALL base+&EB:CALL base+&8:CALL bas
        e+&F1:RETURN                           [1342]
 1500 IF by$="" THEN by=PEEK(base+&6)+256
        *PEEK(base+&7) ELSE by=VAL(by$)+pa     [2CDE]
 1510 CALL base+&3B,by:PRINT"Angehalten b
        ei Adresse: ";HEX$(PEEK(base+&6)+25
        6*PEEK(base+&7)-pa,4):GOTO 90          [0F60]
 1630 INPUT "Datenbits (5-8)";a$:IF a$<>"
        " THEN a=ASC(a$)-48:IF a>4 AND a<9
        THEN dat=a ELSE GOTO 1630              [007A]
 1640 INPUT "Paritaet<3>0,1,2<2>(keine, u
        ngerade, gerade)";a$:IF a$<>"" THEN
         IF ABS(ASC(a$)-49)<2 THEN par=VAL(
        a$) ELSE GOTO 1640                     [0FCE]
 1650 INPUT "Stopbits<3>0,1,2<2>(1, 1<G16
        9>, 2)";a$:IF a$<>"" THEN IF ABS(AS
        C(a$)-49)<2 THEN stbit=VAL(a$) ELSE
          GOTO 1650                            [EA10]
 1660 GOTO 1380                                [7824]
 1700 POKE adr+1,INT(by/256):POKE adr,by-
        INT(by/256)*256:RETURN                 [6222]
```

**Listing 1. DFÜ mit einem eigenen Programm**

```
 15 'fuer VORTEX, AMSTRAD RS232C-Schnitts
    telle                                      [1AE2]
 22 MODE 2:b$="h":bd=300:bd2=300:dat=8:pa
    r=0:hfc=1:stbit=0:csio=&FADD:dsio=&FA
    DC:base=&9C00:h=0:xon$=CHR$(17):xoff$
    =CHR$(19):DIM par$(3):DIM stbit$(3):d
    u$=" Fullduplex":pa=&3000:pg=2^16+bas
    e-1-pa:lf=0                                [3550]
 31 |SETSIO,bd,bd2,hfc,dat,par,stbit           [F9E0]
 70 LOAD"mc.bin",base                          [29D2]
 722 PRINT " Hardware Flow Control ";:IF
     hfc=0 THEN PRINT"aus" ELSE PRINT"an"
                                               [DF4E]
 1380 |SETSIO,bd,bd2,hfc,dat,par,stbit:GO
      TO 90                                    [08A8]
 1600 INPUT "Sendebaudrate ";a$:IF a$<>""
       THEN bd=VAL(a$)                         [B492]
 1610 INPUT "Empfangsrate<2>";a$:IF a$<>"
      " THEN bd2=VAL(a$)                       [8F84]
```

**Listing 2.**

```
 15 'fuer Schneider RS232C-Schnittstelle       [EB8C]
 22 MODE 2:b$="h":bd=300:bd2=300:dat=8:pa
    r=0:stbit=0:dsio=&F8EF:base=&9C00:h=0
    :xon$=CHR$(17):xoff$=CHR$(19):DIM par
    $(3):DIM stbit$(3):du$=" Fullduplex":
    pa=&3000:pg=2^16+base-1-pa:lf=0            [02E0]
 31 '******************  SIO Init ****         [6738]
 32 RESTORE 36                                 [E164]
 33 FOR b=1 TO 12:READ CSIO,A                  [1868]
 34 OUT &F800+csio,a                           [8112]
 35 NEXT                                       [B68E]
 36 DATA &e8,6,&e0,3,&e1,255,&e8,1,&e0,26
    ,&e8,2,&e0,26,&e8,7,&e0,&33,&ec,&88,&
    ed,1,&ee,1                                 [CC56]
 70 LOAD"mc.bin",base+&0                       [86D4]
 1380 OUT &F8E8,7:OUT &F8E0,xbd+xbd2*16:O
      UT &F8EC,(par+1)*2+(stbit+1)*8+(8-d
      at)*32+128:GOTO 90                       [3394]
 1600 INPUT "Sendebaudrate 0,1,2<2>(1200,
       300, 75)";a$:IF a$<>"" THEN IF ABS
      (ASC(a$)-49)<2 THEN a=VAL(a$):xbd=(
      a*2)+1:bd=4^(2-a)*75:ELSE  GOTO 160
      0                                        [8D18]
 1610 INPUT "Empfangsrate 0,1,2<2>(1200,
      300, 75)";a$:IF a$<>"" THEN IF ABS(
      ASC(a$)-49)<2 THEN a=VAL(a$):xbd2=(
      a*2)+1:bd2=4^(2-a)*75:ELSE  GOTO 16
      10                                       [3730]
```

**Listing 3.** [0100]

```
 15 'fuer eigene RS232C-Schnittstellen         [0118]
 22 MODE 2:b$="h":bd=300:bd2=300:dat=8:pa
    r=0:stbit=0:dsio=&F8E4:csio=&F8E5:ctc
    =&F8E8:base=&9C00:h=0:xon$=CHR$(17):x
    off$=CHR$(19):DIM par$(3):DIM stbit$(
    3):du$=" Fullduplex":pa=&3000:pg=2^16
    +base-1-pa:lf=0                            [F6E6]
 31 '******************  SIO Init ****         [6738]
 32 RESTORE 36                                 [E164]
 33 FOR b=1 TO 9:READ a                        [B4A0]
 34 OUT csio,a                                 [5DB4]
 35 NEXT                                       [B68E]
 36 DATA &18,4,&c4,3,&c1,5,&68,1,0             [4D74]
 40 '******************  CTC Init ****         [6F16]
 41 RESTORE 45                                 [0D64]
 42 FOR b=1 TO 0:READ ctc,a                    [795A]
 43 OUT ctc,a                                  [76CC]
 44 NEXT                                       [C58E]
 45 DATA                                       [B446]
 70 LOAD"mc.bin",base+&0                       [86D4]
 1380 OUT csio,4:OUT csio,&40+par-(par>1)
      +(stbit+1)*4                             [D284]
 1381 OUT csio,5:OUT csio,(dat-5)*32+8         [A63C]
 1382 OUT csio,3:OUT csio,(dat-5)*64+1         [DC36]
 1383 OUT ctc,7:OUT ctc,4^xdb*13               [4174]
 1384 OUT ctc+1,7:OUT ctc+1,4^xdb2*13:GOT
      O 90                                     [2F42]
 1600 INPUT "Sendebaudrate 0,1,2<2>(1200,
       300, 75)";a$:IF a$<>"" THEN IF ABS
      (ASC(a$)-49)<2 THEN a=VAL(a$):xbd=(
      a*2)+1:bd=4^(2-a)*75:ELSE  GOTO 160
      0                                        [8D18]
 1610 INPUT "Empfangsrate 0,1,2<2>(1200,
      300, 75)";a$:IF a$<>"" THEN IF ABS(
      ASC(a$)-49)<2 THEN a=VAL(a$):xbd2=(
      a*2)+1:bd2=4^(2-a)*75:ELSE  GOTO 16
      10                                       [3730]
```

**Listing 4.**

```
100  '********************************         [A284]
101  '* 1V.DAT - DATA-Lader von 'CPC' *         [929C]
102  '********************************         [1C88]
103  '                                         [DEB6]
104 DATA 4000,00,30,FF,9B,00,30,00,00,1A90     [99CE]
105 DATA 4008,CD,09,BB,30,06,FE,09,C8,73B2     [49A4]
106 DATA 4010,CD,FB,9C,CD,FD,9C,18,F0,4248     [6516]
107 DATA 4018,CD,80,BC,D0,CD,FB,9C,18,58A4     [DB12]
108 DATA 4020,E7,2A,00,9C,CD,9B,9C,CD,7531     [57CA]
109 DATA 4028,95,BC,23,22,00,C0,ED,4B,6151     [7352]
110 DATA 4030,04,9C,7C,B8,20,EE,7D,B9,23FB     [BBC4]
111 DATA 4038,20,EA,C9,DD,6E,00,DD,66,3CDC     [2F00]
112 DATA 4040,01,CD,09,BB,30,09,CD,06,3968     [AC54]
113 DATA 4048,BB,FE,09,22,06,9C,C8,CD,621D     [35D0]
114 DATA 4050,9B,9C,23,CD,58,9C,18,E9,6259     [EF9C]
115 DATA 4058,E6,FF,FE,0D,20,05,CD,5A,5304     [74C8]
116 DATA 4060,BB,3E,0A,FE,20,D2,5A,BB,5EE7     [F810]
117 DATA 4068,E5,21,73,9C,85,6F,7E,E1,78E9     [79B8]
118 DATA 4070,C3,5A,BB,00,01,00,03,00,606E     [8200]
119 DATA 4078,00,06,07,08,09,0A,0B,0C,019A     [8806]
120 DATA 4080,0D,00,00,10,00,00,00,00,0780     [4952]
121 DATA 4088,00,00,00,00,00,00,00,00,0000     [EE1C]
122 DATA 4090,00,00,00,00,00,00,00,00,0000     [2310]
123 DATA 4098,00,00,00,E5,CD,F7,9C,7E,0AA2     [1196]
124 DATA 40A0,CD,F9,9C,E1,C9,F5,C5,E5,41A3     [8618]
125 DATA 40A8,2A,04,9C,CD,F7,9C,77,CD,0EBB     [CF38]
126 DATA 40B0,F9,9C,23,22,04,9C,3F,ED,5F03     [C0B2]
127 DATA 40B8,4B,02,9C,ED,42,7C,B5,CC,3A16     [50E6]
128 DATA 40C0,C6,9C,E1,C1,F1,C9,CD,F1,51F7     [1F16]
129 DATA 40C8,9C,CD,21,9C,CD,EB,9C,2A,7476     [E330]
130 DATA 40D0,00,9C,22,04,9C,C9,2A,04,2494     [8A30]
131 DATA 40D8,9C,CD,80,BC,30,09,CD,A5,661B     [F6E6]
132 DATA 40E0,9C,23,22,00,C0,18,F2,22,4526     [3906]
133 DATA 40E8,04,9C,C9,3E,11,CD,FB,9C,3D16     [0CFA]
134 DATA 40F0,C9,3E,13,CD,FB,9C,C9,C9,6143     [D102]
135 DATA 40F8,00,C9,00,18,02,18,13,F5,3363     [2AFE]
140 DATA 4120,9C,CD,58,9C,C9,00,00,00,79C8     [9D6A]
141 DATA *ENDE*                                [89C0]
142 adr=&4000:zeile=104:MEMORY adr-1           [B904]
143 READ d$:IF d$="*ENDE*"THEN 154             [248E]
144 pr=0                                       [4B10]
145 FOR i=1 TO 8                               [0266]
146 READ a$:a=VAL("&"+a$)                      [2944]
147 POKE adr,a:adr=adr+1                       [D620]
148 pr=pr*2:IF pr>65535 THEN pr=pr-65535       [5BA0]
149 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                          [52B8]
150 NEXT i                                     [7BFC]
151 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                             [BD88]
152 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                      [3010]
153 zeile=zeile+1:GOTO 143                     [DD5E]
154 SAVE"mc.bin",B,&4000,&4130:END             [8026]
```

**Listing 5.**

```
136 DATA 4100,01,DD,FA,ED,78,E6,04,28,2628     [AE90]
137 DATA 4108,FA,0E,DC,F1,ED,79,CD,21,6D27     [9F1C]
138 DATA 4110,9D,C9,01,DD,FA,ED,78,E6,7542     [9FEC]
139 DATA 4118,01,C8,0E,DC,ED,78,CD,A5,3937     [10E8]
```

**Listing 6.**

```
136 DATA 4100,01,E5,F8,ED,78,E6,04,28,2868     [ED6E]
137 DATA 4108,FA,0E,F4,F1,ED,79,CD,21,6827     [80EA]
138 DATA 4110,9D,C9,01,E5,F8,ED,78,E6,76D2     [DAE0]
139 DATA 4118,01,C8,0E,F4,ED,78,CD,A5,3BB7     [60FE]
```

**Listing 7.**

```
136 DATA 4100,01,EE,F8,ED,78,E6,80,28,2BA0     [E6B0]
137 DATA 4108,FA,0E,EF,F1,ED,79,CD,21,6B47     [BC24]
138 DATA 4110,9D,C9,01,ED,F8,ED,78,E6,7652     [6EE0]
139 DATA 4118,80,C8,0E,EF,ED,78,CD,A5,7A87     [1920]
```

**Listing 8.**

# Spritzige Sprites

**Nur wenige Bytes Maschinencode erlauben extrem schnelle Bildschirmausgaben. Da haben sogar die Sprites des Commodore 64 das Nachsehen.**

Vorsicht – Feind von rechts! Das kleine Männchen rast mit ungeheuerlicher Geschwindigkeit auf unsern Held in der Mitte des Bildschirms zu. Und dabei nähern sich von oben noch zwei fliegende Untertassen – ganz zu schweigen von den Wogen des turkistanischen Meers auf Xylon.

Je mehr Bewegung auf dem Bildschirm, desto besser ist ein Spiel. Software wirkt um Längen professioneller, wenn die Bildausgabe nur Bruchteile einer Sekunde dauert. Bei den Schneider-Computern der CPC-Serie wird der Bildschirm bei der Zeichenausgabe mit sehr komplexen und leistungsfähigen Betriebssystemroutinen angesprochen. Diese sind dadurch natürlich sehr lang – und dementsprechend langsam. Die meisten Konkurrenzprodukte unterscheiden zwischen einem Text- und einem Grafikmodus. Bei Schneider gibt es nur den Grafikmodus. Jeder einzelne Buchstabe und jedes Grafikzeichen wird als Punktmuster in den Bildschirm geschrieben. Dabei wird unter anderem auf Windows und Farben geachtet, zwischen zu druckenden und Steuerzeichen unterschieden und was der Dinge mehr sind. Ferner sind alle Routinen so geschrieben, daß sie in allen Bildschirmmodi gleichermaßen arbeiten.

Alle Programme, die eine direkte Ansteuerung des Bildschirms verlangen, laufen deshalb auf einem Schneider-Computer sehr langsam ab. Wer sich bereits einmal daran versucht hat, ein Spiel mit viel Bewegung auf dem Schneider zu realisieren, der weiß davon ein Lied zu singen. Ganz schlimm wird es, wenn Sie versuchen, Zeichen direkt vom Bildschirm zu lesen (beispielsweise beim 664/6128 mit COPYCHR$). Mit den hier vorgestellten Sprite-Routinen lösen sich viele Programmierprobleme wie von selbst.

Zunächst wird das verwendete Sprite-Konzept vorgestellt. Ein Sprite ist ein 8x8 Punkte großer Bildschirmbereich und entspricht damit im Modus 1 genau einem Zeichen. Da in diesem Modus jeder Punkt auf dem Bildschirm durch zwei Bit dargestellt wird, braucht man für ein Sprite 128 (=8x8x2) Bit Speicherplatz (Bild 1). Die gleiche Struktur haben alle Buchstaben, die im Modus 1 auf den Bildschirm gebracht werden. Das Betriebssystem geht dabei von einer 8x8-Schablone für jedes Zeichen aus. Diese Schablone wird je nach Schriftfarbe in das entsprechende Bit-Muster für die Bildschirmausgabe umgerechnet. Zudem wird aber jedesmal auch noch die effektive Adresse des Zeichens im Video-RAM aus der X- und Y-Position des Cursors bestimmt. Das kostet ebenfalls viel Rechenzeit.

Bei unseren Sprite-Routinen verzichten wir auf alle diese rechenintensiven Vorgänge. Die Anweisung erwartet das fertige Bitmuster aus 16 Byte. Diese werden mit einer sehr kurzen und daher extrem schnellen Routine direkt in den Bildspeicher geschrieben.

Im Modus 1 gibt es 1000 Zeichen auf dem Bildschirm (40x25), somit auch 1000 erlaubte Sprite-Positionen. Sprites können also nur an solche Positionen geschrieben werden, an denen bei der normalen Textausgabe Buchstaben erscheinen. In der Praxis ist das nur ein kleiner Nachteil.

Die Steuerung der Sprites erfolgt mit den zwei Kommandos PUT und GET. Beide sind sehr ähnlich aufgebaut. Listing 1 zeigt den Assembler-Quellcode der beiden Routinen. Listing 2 ist das gleiche Programm als Basic-Lader. Das Maschinencode-Programm ist voll relokatibel, das heißt, es darf unverändert an jede beliebige RAM-Adresse gestellt werden. Bewußt wurden PUT und GET nicht als RSX-Anweisungen konzipiert. RSX-Aufrufe sind vergleichsweise langsam, da das Basic bei jedem RSX zuerst alle angeschlossenen ROMs nach dem Befehl durch-

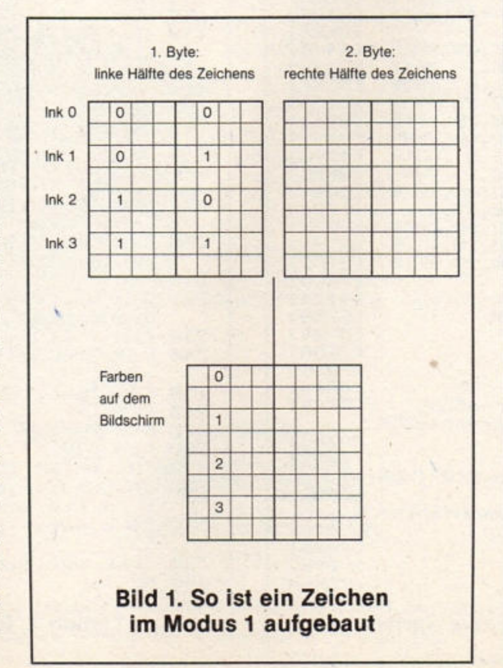



**Bild 1. So ist ein Zeichen im Modus 1 aufgebaut**

```
; ------------
; S P R I T E S
; ------------
;
ORG     #A000
;voll relokatibel !
;
;
;Sprungvektoren
;
JR      PUT
JR      GET
;
;
;Sprite von Bildschirm lesen
;Aufruf: CALL GET,ScreenAdr,@Sprite$
;
GET:CP  2;zwei Parameter?
RET     NZ;nein, zurueck
LD      H,(IX+1);Stringdescr.
LD      L,(IX+0)
LD      A,(HL)  ;Laenge
CP      16      ;kein Sprite?
RET     NZ;dann zurueck
INC     HL
LD      E,(HL)  ;Spriteadresse
INC     HL
LD      D,(HL)
LD      H,(IX+3);ScreenAdr
LD      L,(IX+2)
ADD     HL,HL;effektive Adr.
LD      A,H
AND     7;Korrektur
OR      #C0;Video-RAM
LD      H,A
LD      BC,#800;Zeilendiff.
LD      A,8;Zaehler
GET1:LDI        ;zwei Byte
LDI     ;holen
ADD     HL,BC;naechste Zeile
INC     BC;BC korrigieren
INC     BC
DEC     A;Zeichen fertig
JR      NZ,GET1;nein, noch nicht
RET             ;aber jetzt
;
;
;Sprite auf Bildschirm ausgeben
;Aufruf: CALL PUT,ScreenAdr,@Sprite$
;
PUT:CP  2;zwei Parameter?
RET     NZ;nein
LD      H,(IX+1)        ;Stringdescr.
LD      L,(IX+0)
LD      A,(HL);Laenge
CP      16      ;Sprite?
RET     NZ;nein, zurueck
INC     HL
LD      E,(HL)  ;Spriteadr.
INC     HL
LD      D,(HL)
LD      H,(IX+3)        ;ScreenAdr
LD      L,(IX+2)
ADD     HL,HL;effektive
LD      A,H;Videoadresse
AND     7;berechnen
OR      #C0;Video-RAM
LD      H,A
LD      BC,#800;Zeilendiff.
LD      A,8;Zaehler
PUT1:EX DE,HL
LDI;zwei Byte
LDI;schreiben
EX      DE,HL
ADD     HL,BC;naechste Zeile
INC     BC;BC korrigieren
INC     BC
DEC     A;fertig?
JR      NZ,PUT1;noch nicht
RET             ;OK
```

**Listing 1. Der Assemblercode für GET und PUT**

sucht. Unser Aufruf soll aber möglichst wenig Zeit brauchen. Der Aufruf über CALL ist genauso komfortabel – vorausgesetzt, man definiert gleich zu Anfang die Variablen PUT und GET als Startadresse beziehungsweise Startadresse plus 2.

Um vom Basic-Programm aus bequem mit den Sprites zu arbeiten, holen sich die Maschinencode-Programme PUT und GET die Sprite-Daten aus dem Speicherbereich der Stringvariablen. Mit anderen Worten: Sprites werden in Strings gespeichert. Jeder Sprite-String muß exakt eine Länge von 16 Byte haben. Solch ein String muß deshalb von dem Programm definiert werden, bevor es zum Aufruf der Sprite-Routinen kommt. Da die Information für die Sprites reine Bit-Muster sind, lassen sich solche Sprite-Strings mit der PRINT-Anweisung nicht auf dem Bildschirm ausgeben.

Beim Aufruf von PUT und GET müssen die Ausgabeadresse und die Adresse des Sprite-Strings angegeben werden. Die Ausgabeadresse ist einfach die Nummer des Zeichens auf dem Bildschirm im Modus 1 (Bild 2). Die Adresse des Sprite-Strings wird durch den Operator »@« gewonnen. Um beispielsweise das Sprite X$ an den Anfang der zweiten Bildschirmzeile (also an Position 40) zu schreiben, müssen Sie in Ihr Programm »CALL PUT,40,@X$« schreiben. Um das erste Zeichen der letzten Zeile aus dem Bildschirm in einen Sprite-String mit dem Namen HILF$ zu lesen, geben Sie »CALL GET,960,@HILF$« ein.

Ein paar Dinge beachten Sie bitte bei der Arbeit mit den Sprites:

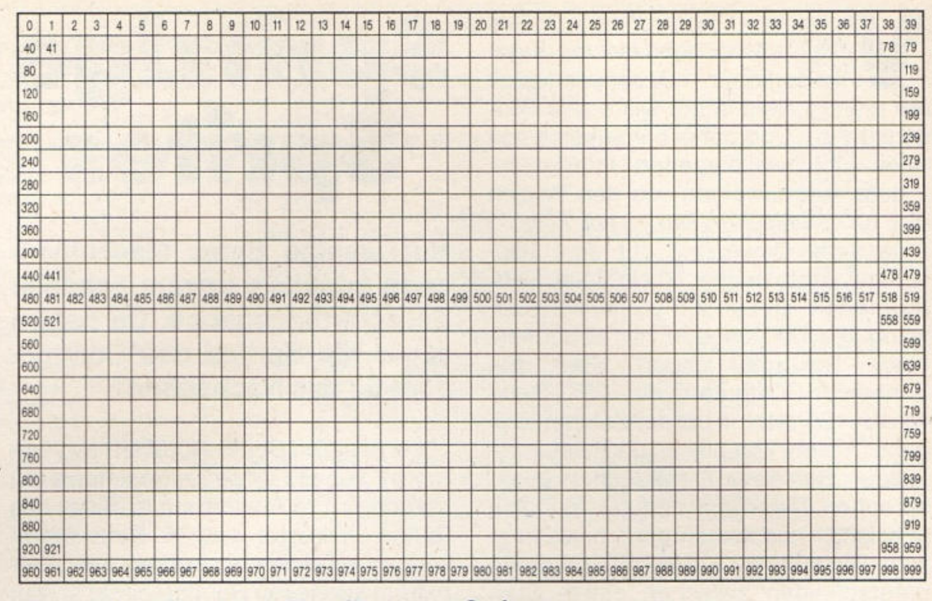



**Bild 2. 1000 Bildschirmplätze für unsere Sprites**

- Der Bildschirm darf nicht gescrollt werden
- Wenn die PUT- oder GET-Befehle fehlerhaft sind, wird die Sprite-Routine ohne irgendwelche Aktionen beendet.
- Die Sprite-Routinen sind für den Bildschirmmodus 1 ausgelegt.

Im Listing 3 finden Sie ein Unterprogramm in Basic, das Sprites aus DATA-Zeilen erzeugt. Die ersten Zeilen zeigen dabei gleich ein Demonstrationsprogramm zum Einsatz von Sprites. Der Sprite-Generator wird mit »GOSUB 60000« aufgerufen, nachdem der DATA-Zeiger mittels »RESTORE« auf die gewünschte Zeile gesetzt wurde. Jeweils 8 DATA-Zeilen zu je 8 Zeichen definieren ein Sprite. Die Ziffern 0, 1, 2 und 3 innerhalb einer solchen Zeile stehen für die vier im Modus 1 erlaubten Farben. Statt der Null ist auch ein Leerzeichen (Space) zulässig. Die Basic-Routine erzeugt aus den DATA-Angaben die Sprite-Binärwerte und speichert den entsprechenden Sprite in der Variablen SPRITE$.

Da sich mit GET nicht nur Original-Sprites, sondern beliebige Zeichen aus dem Bildschirm in einen Sprite-String einlesen lassen, ist es sehr einfach, jedes Zeichen des Schneider-Zeichensatzes in ein Sprite zu verwandeln. Hierzu dient beispielsweise das Programm aus Listing 4.

(Anne Everts/hg)

```
100 '************************************  [31D4]
101 '* SPRITE.DAT - DATA-Lader von 'CPC' *  [0F3C]
102 '************************************  [A3D8]
103 '                                       [DEB6]
104 DATA A000,18,30,18,00,FE,02,C0,DD,05A5  [A448]
105 DATA A008,66,01,DD,6E,00,7E,FE,10,2E14  [D592]
106 DATA A010,C0,23,5E,23,56,DD,66,03,603B  [5346]
107 DATA A018,DD,6E,02,29,7C,E6,07,F6,7750  [CCA6]
108 DATA A020,C0,67,01,00,08,3E,08,ED,79A5  [8B54]
109 DATA A028,A0,ED,A0,09,03,03,3D,20,7F9E  [9492]
110 DATA A030,F6,C9,FE,02,C0,DD,66,01,5319  [4996]
111 DATA A038,DD,6E,00,7E,FE,10,C0,23,74F3  [8EB8]
112 DATA A040,5E,23,56,DD,66,03,DD,6E,2238  [C284]
113 DATA A048,02,29,7C,E6,07,F6,C0,67,08A7  [FE7A]
114 DATA A050,01,00,08,3E,08,EB,ED,A0,00F6  [EE70]
115 DATA A058,ED,A0,EB,09,03,03,3D,20,433E  [8586]
116 DATA A060,F4,C9,00,53,50,20,20,20,4F10  [81F2]
117 DATA *ENDE*                             [74C6]
118 adr=&A000:zeile=104:MEMORY adr-1        [4124]
119 READ d$:IF d$="*ENDE*"THEN 130          [E288]
120 pr=0                                    [5F04]
121 FOR i=1 TO 8                            [2E5A]
122 READ a$:a=VAL("&"+a$)                   [FB38]
123 POKE adr,a:adr=adr+1                    [DA14]
124 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [DF94]
125 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                       [A2AC]
126 NEXT i                                  [4402]
127 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                          [308E]
128 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                   [5A16]
129 zeile=zeile+1:GOTO 119                  [1A6A]
130 SAVE"SPRITE.BIN",B,&A000,&61            [C09E]
131 PRINT d$:END                            [73F6]
```

**Listing 2. Der Basic-Lader für unsere Sprites**

```
100 REM Sprite-Demo                         [E5BC]
110 REM -----------                         [724A]
120 REM                                     [5C2E]
130 IF PEEK(&A000)<>24 THEN MEMORY &9FFF
    :LOAD"sprite.bin",&A000                 [F010]
140 DEFINT a-z:put=&A000:get=&A002:MODE
    1:z$="bitte warten !":PRINT z$          [214C]
150 REM                                     [5D34]
160 REM Sprites erzeugen                    [5A54]
170 REM                                     [6B38]
180 RESTORE 1010:GOSUB 60000:x1$=sprite$
    :PRINT TAB(5);z$                        [DA56]
190 RESTORE 1120:GOSUB 60000:x2$=sprite$
    :PRINT TAB(10);z$                       [5FB6]
200 RESTORE 1230:GOSUB 60000:x3$=sprite$
    :PRINT TAB(15);z$                       [B5B6]
210 REM                                     [652E]
220 REM Hilfs-Strings zum Lesen des Hint
    ergrundes in Sprites definieren         [57E0]
230 REM                                     [6332]
240 h1$=SPACE$(16):h2$=h1$:h3$=h2$:delay
    =10                                     [8E8A]
250 CALL get,120,@h1$:CALL get,121,@h2$     [1676]
260 REM                                     [5A38]
270 REM Sprite nach rechts bewegen          [9228]
280 REM                                     [783C]
290 FOR i=120 TO 158                        [C6F8]
300 CALL put,i,@x1$:CALL put,i+1,@x3$:RE
    M Sprite schreiben                      [CC32]
310 CALL get,i+2,@h3$:FOR k=1 TO delay:N
    EXT                                     [0CA4]
320 CALL put,i,@h1$:h1$=h2$:h2$=h3$         [54DC]
330 NEXT                                    [57EA]
```

**Listing 3. Eine Demonstration der Sprites**

```
340 CALL get,i-2,@h3$                          [B5A4]
350 REM                                        [6D38]
360 REM Sprite nach links bewegen              [1258]
370 REM                                        [5F3C]
380 FOR i=158 TO 120 STEP-1                    [686C]
390 CALL put,i,@x3$:CALL put,i+1,@x2$:RE
    M Sprite schreiben                         [1C46]
400 CALL get,i-1,@h3$:FOR k=1 TO delay:N
    EXT                                        [68A6]
410 CALL put,i+1,@h1$:h1$=h2$:h2$=h3$          [7294]
420 NEXT                                       [D2EA]
430 GOTO 290                                   [6656]
1000 REM                                       [288A]
1010 REM Sprite x1$                            [A394]
1020 REM                                       [1A8E]
1030 DATA "1<3>11<2>"                          [1D2A]
1040 DATA "11<2>11<2>"                         [CF4E]
1050 DATA "311111<2>"                          [8D98]
1060 DATA "33111111"                           [25E2]
1070 DATA "33111111"                           [2CE4]
1080 DATA "311111<2>"                          [A09E]
1090 DATA "11<2>11<2>"                         [DE58]
1100 DATA "1<3>11<2>"                          [1C26]
1110 REM                                       [198E]
1120 REM Sprite x2$                            [9F9A]
1130 REM                                       [2792]
1140 DATA "<2>11<3>1"                          [FD2E]
1150 DATA "<2>11<2>11"                         [6452]
1160 DATA "<2>11 113"                          [BC7A]
1170 DATA "11111133"                           [DAE6]
1180 DATA "11111133"                           [8FE8]
1190 DATA "<2>11 113"                          [3D80]
1200 DATA "<2>11<2>11"                         [674A]
1210 DATA "<2>11<3>1"                          [002A]
1220 REM                                       [3E92]
1230 REM Sprite x3$                            [87A0]
1240 REM                                       [1C96]
1250 DATA "<8>"                                [D7CC]
1260 DATA "<2>1111<2>"                         [1956]
1270 DATA " 111111 "                           [7B9C]
1280 DATA "11111111"                           [C1E2]
1290 DATA "11111111"                           [40E4]
1300 DATA " 111111 "                           [7A90]
1310 DATA "<2>1111<2>"                         [184E]
1320 DATA "<8>"                                [D6C8]
59990 REM SPRITEMAKER                          [22F6]
59991 REM ------------                         [7948]
59992 REM Unterprogramm zum erzeugen von
       Sprites aus Datazeilen                  [F462]
59993 REM Aufruf: RESTORE nn:GOSUB 60000
                                               [9FD4]
59994 REM kehrt zurueck mit Spritedaten
      in Sprite$                               [7F7C]
59995 REM intern benutzte Variablen:           [F290]
59996 REM   x$,r$,b1$,b2$,ch$,i,j,k            [C4F8]
59997 REM                                      [BE36]
60000 sprite$="":FOR k=1 TO 8:READ r$:x$
      =BIN$(0,16):j=1                          [F692]
60010 FOR i=1 TO 8:ch$=MID$(r$,i,1):IF c
      h$=" " THEN ch$="0"                      [C8AA]
60020 IF ch$<"0" OR ch$>"3" THEN ERROR 5
                                               [C172]
60030 b1$=BIN$(VAL(ch$),2):b2$=RIGHT$(b1
      $,1):b1$=LEFT$(b1$,1)                    [40E8]
60040 MID$(x$,j,1)=b1$:MID$(x$,j+4,1)=b2
      $:j=j+1:IF j=5 THEN j=9                  [9E26]
60050 NEXT:sprite$=sprite$+CHR$(VAL("&X"
      +LEFT$(x$,8)))+CHR$(VAL("&X"+RIGHT
      $(x$,8))):NEXT                           [D01E]
60060 RETURN                                   [5CF8]
```

**Listing 3. Eine Demonstration der Sprites (Schluß)**

```
10 IF PEEK(&9000) <> 24 THEN MEMORY &8FF
   F:LOAD"SPRITE.BIN",&9000                    [74C8]
20 DEFINT A-Z:PUT=&9000:GET=PUT+2              [D360]
30 INPUT "Zeichen-Nummer:",ZN                  [FDD8]
40 MODE 1:PRINT CHR$(ZN);:REM Zeichen ob
   en links einschreiben                       [DA5C]
50 SPR$=SPACE$(16):REM 16 Bytes langen S
   tring erzeugen                              [CD40]
60 CALL GET,0,@SPR$:REM Sprite einlesen        [F656]
70 FOR I=1 TO 999:REM Beispiel...              [F790]
80 CALL PUT,I,@SPR$                            [266E]
90 NEXT                                        [C590]
```

**Listing 4. Jedes Zeichen ist ein Sprite**





# Der ewige Wettstreit

**Ob Sie den Schriftverkehr Ihrer Firma abwickeln wollen, oder nur gelegentlich einen privaten Brief schreiben, stellt ganz unterschiedliche Anforderungen an eine Textverarbeitung. Wir sagen Ihnen was die wichtigsten Programme leisten.**

Die Be- und Verarbeitung von Texten ist eine klassische Anwendung für Computer. So ist es kein Wunder, daß die Softwarehäuser viel Zeit und Geld in dieses Gebiet investieren, um leistungsfähige Textverarbeitungsprogramme zu entwikkeln. Wir vergleichen für Sie aus diesem Angebot fünf Programme, die eine bedeutende Marktstellung erreicht haben. Hier die Kandidaten kurz vorgestellt:

»Wordstar« erreichte einst auf Personal Computern (damals noch 8-Bit-Maschinen) die weiteste Verbreitung. Es ist berühmt für seine hohe Leistungsfähigkeit und berüchtigt für seine komplizierte Bedienung.

»Tasword« von Tasman ist das Wordstar des kleinen Mannes mit Heimcomputer. In der Bedienung ähnelt es Wordstar, eine Befehlskompatibilität besteht allerdings nicht.

Das Textverarbeitungssystem »Star-Texter« des Sybex-Verlags verwendet eine Mischung aus Menüs und Tastenbefehlen. Es tut sich durch einige praktische Besonderheiten hervor.

»Star-Writer« stammt von Star-Division und versucht, Pull-down-Menüs und Fenster in die Textverarbeitung zu integrieren. Der Versuch ist zwar lobenswert, kränkelt jedoch an der Verarbeitungsgeschwindigkeit des CPC.

Das Programm »Textomat« von Data-Becker setzt konsequent auf Menüsteuerung. Dies hat zwar für Neulinge den Vorteil einer einfachen Bedienung, »alte Hasen« hingegen hält es auf.

## Wordstar antiquiert?

Wordstar ist der Oldie unter den Textverarbeitungsprogrammen. Es ist schon seit einigen Jahren auf dem Markt. Es stellt sich die Frage, ob es mit modernen Systemen noch mithalten kann.

Den amerikanischen Hersteller Micro-Pro stellen jedoch die Verkaufszahlen von Wordstar immer noch zufrieden. Mehr als eine Million Exemplare wurden insgesamt verkauft. Ein Vielfaches dieser Zahl ist zudem als Raubkopie unterwegs, da Wordstar keinen Kopierschutz besitzt.

Vor 1 1/2 Jahren kostete Wordstar noch über 1000 Mark. Jetzt wird es in Versionen für die Schneider CPCs, schon für 199 Mark angeboten. Diese (überfällige) Preiskorrektur ist natürlich für den Käufer sehr angenehm und verhilft dem Programm auch unter den Heimanwendern zu einer beachtlichen Popularität.

Wordstar arbeitet unter CP/M. Dieses Betriebssystem erlebt seit einiger Zeit eine neue Blüte, denn Schneider liefert es kostenlos zum ersten Laufwerk für den CPC 464, und im Lieferumfang des CPC 664/6128 ist es sogar mit enthalten. CP/M ist weitgehend genormt. Dies bezieht sich jedoch nicht auf das Diskettenformat und die Wege zur Bildschirmansteuerung. So muß sich der Käufer bei Wordstar zwischen der Anpassung auf CP/M 2.2 für den CPC 464/664 und auf CP/M Plus für den CPC 6128 entscheiden. Außerdem wird Wordstar auf 3-Zoll-Disketten und im Vortex-Format (5 1/4 Zoll) zu jeweils 199 Mark geliefert.

Was läßt sich nun konkret mit Wordstar anstellen? Die Antwort von Wordstar-Fans: Praktisch alles! Es gibt kein anderes Textverarbeitungssystem für 8-Bit-Computer, das in der Leistungsfähigkeit auch nur annähernd an Wordstar heranreicht.

Neben der Cursorsteuerung über die Pfeiltasten können Sie mit <CTRL+W> und <CTRL+Z> den Bildschirminhalt unter dem Cursor durchlaufen lassen. Seitenweises Blättern ist ebenso möglich wie der Sprung an Dateianfang und Dateiende. Weitere Zielpunkte des Cursors sind Blockanfang, Blockende, Zeilenanfang und Zeilenende, die oberste und unterste Bildschirmzeile, der nächste Tabulatorstop sowie das Wort links und rechts des Cursors.

Die Eigenschaften, die alle Textverarbeitungsprogramme beherrschen, zum Beispiel das Einfügen und Löschen von Zeilen, Block- und Dateibefehle, sind für Wordstar selbstverständlich. Interessant sind Funktionen wie Wortumbruch am Zeilenende (auf Wunsch abschaltbar), wählbarer Blocksatz, der auch auf dem Bildschirm dargestellt wird, und die automatische Trennhilfe für Wörter am Zeilenende. Diese Trennhilfe funktioniert trotz des amerikanischen Algorithmus erstaunlich gut – Treffsicherheit bis zu 90 Prozent!

Sobald es ans Ausdrucken von Texten geht, zeigt Wordstar auch in diesem Bereich seine Fähigkeiten. Es kennt Druckerbefehle, die direkt in den Text geschrieben werden. So legt ».PL« die Seitenlänge fest, und ».PA« bewirkt einen neuen Seitenanfang. Die Druckroutine läßt am oberen und unteren Rand eines Blattes drei Zeilen frei. Mit den Punktbefehlen ».MT« (Margin At Top) und ».MB« (Margin At Bottom) können Sie die Größe des Randes verändern.

## Mit dem Drucker per Du

Tippen Sie größere Texte wie Berichte, Doktorarbeiten oder lange Briefe ein, werden Sie auf eine Seitennumerierung nicht verzichten wollen. Wordstar bietet Kopf- oder Fußzeilen mit blanken Seitennummern oder Nummern in Texte eingebettet (beispielsweise »- Seite 23 -«). Wenn Sie ein Buch schreiben wollen, druckt Wordstar die Seitenzahlen abwechselnd an den linken oder rechten Rand.

Über die Leistungsfähigkeit von Wordstar ließe sich noch viel schreiben, doch es stellt sich die Frage, ob Wordstar bei dieser Funktionsvielfalt noch einfach zu bedienen ist. Und hier zeigt sich die andere Seite der Medaille: Wordstar hat den zweifelhaften Ruf, eines der schwierigsten Textprogramme auf dem Markt zu sein. Auch wenn das etwas übertrieben ist, steckt darin doch ein wahrer Kern. Viele Funktionen bedeutet natürlich viele Tastenkombinationen, um diese Funktionen aufzurufen. Viele der Tastenkombinationen sind aber einfach kryptisch. Warum rückt zum Beispiel <CTRL+O> + <G> einen Absatz ein? So liegen viele Funktionen von Wordstar brach, weil der Benutzer nicht weiß, wie er sie aufrufen soll, und im Handbuch will man natürlich auch nicht dauernd blättern.

MicroPro ist sich dieses Problems durchaus bewußt. Deshalb bietet Wordstar Hilfsbildschirme an, die in die obere Bildschirmhälfte eingeblendet werden und über die einzelnen Kommandos einer Befehlsgruppe Auskunft geben. Beim Programmstart nimmt das Hilfsmenü etwa die Hälfte des Bildschirms ein. Weil dadurch aber der Editierbild-

schirm recht klein ist, werden die meisten Anwender nach einer kurzen Einarbeitungszeit dieses Hauptmenü ausblenden. In der zweiten Hilfsstufe zeigt Wordstar nur noch Menüs zu den Unterfunktionen an, und dies auch nur, wenn der Benutzer bei der Befehlseingabe zu lange wartet. So paßt sich Wordstar automatisch dem Wissensstand an.

Bei den Wordstar-Versionen für die drei CPC-Modelle ist die CP/M-Plus-Version besser gelungen. So greift beispielsweise das Scrolling auf die leistungsfähigen Bildschirm-Steuerzeichen zurück, die wiederum direkt das Betriebssystem aufrufen. Natürlich kann man auch mit der CP/M-2.2-Version sinnvoll arbeiten, nur geht eben alles deutlich langsamer.

Wenn man die enorme Leistungsfähigkeit von Wordstar betrachtet, stellt sich unweigerlich die Frage, wie ein so riesiges Programm im Speicher des CPC untergebracht werden kann und gleichzeitig noch genügend Platz für den Text bleibt.

Wordstar besteht aus drei Programmteilen, die zusammen eine Länge von 75 KByte haben. Eigentlich dürfte da kein einziges Byte für den zu bearbeitenden Text frei bleiben. Doch hier kommt das Disketten-Laufwerk ins Spiel. Wordstar arbeitet nämlich mit Overlays. Das sind Programmteile, die nur bei Bedarf von der Diskette in den Speicher geladen werden. Im Speicher verbleibt nur ein »Rumpfprogramm« mit 16 KByte Länge. Wenn Sie eine Programmfunktion aufrufen, die selten benutzt wird, lädt Wordstar ein Overlay und führt die Funktion aus.

## Speichertricks mit Overlay

So bleibt ein großer Teil des Speichers frei. Allerdings ist der freie Speicherplatz (TPA) unter CP/M 2.2 auf dem CPC mit rund 39 KByte recht knapp bemessen. Allzu lange Texte ließen sich damit nicht bearbeiten, wenn Wordstar nicht noch eine weitere Finesse besitzen würde: Das Programm legt die eingegebenen Texte nur vorübergehend im Arbeitsspeicher ab. Sobald der Speicher voll ist, sichert Wordstar automatisch den Text auf Diskette und leert den Hauptspeicher. Wenn Sie den Cursor durch den Text bewegen, lädt das Programm die jeweils benötigten Teile des Textes nach. Aufgrund dieser Technik wird die Textlänge nur durch die Diskettenkapazität begrenzt. Bei der 61KByte-TPA unter CP/M Plus ist der durch das Speichern entstehende Geschwindigkeitsverlust geringer, da hier im Hauptspeicher mehr Platz für den Text übrig bleibt.

Zwei Einschränkungen betreffen die CP/M-2.2-Version von Wordstar: Aufgrund des kleinen Speichers sind Blockverschiebe-Operationen nur bis zu einer Länge von einer Textzeile möglich. Ein notdürftiger Ausweg besteht darin, den Block auf die Diskette zu schreiben und an der Zielstelle wieder einzulesen.

Die zweite Beschränkung betrifft das Ausdrucken von Dateien während der Bearbeitung von Texten. Diese sehr nützliche Eigenschaft kann der Benutzer bei der kleinen TPA nicht aufrufen. Sobald der CPC jedoch mit einer Speichererweiterungskarte von Vortex aufgerüstet wird, fallen diese beiden Einschränkungen weg, und Wordstar schwingt sich zu ungeahnten Geschwindigkeiten auf.

Weil sämtliche Programmteile von Wordstar ständig auf der Arbeitsdiskette stehen müssen, bleiben unter AMSDOS nur noch 94 KByte Diskettenkapazität frei. Wordstar legt zusätzlich eine Arbeitskopie und eine EDBACKUP-Datei an, so daß die 94KByte noch durch 3 geteilt werden müssen. Bei Verwendung von 3-Zoll-Disketten mit 360 KByte Speicherkapazität ist die Textlänge folglich auf 30000 Zeichen beschränkt. Ändern können Sie die Situation durch Benutzung eines zweiten Laufwerks, einer RAM-Disk oder über Laufwerke mit erhöhter Speicherkapazität.

Für wen ist Wordstar geeignet? Bestimmt nicht für denjenigen, der auf großen Bedienungskomfort Wert legt. Aber garantiert für jeden, der lange Texte bearbeiten muß und eine Vielzahl von Editiermöglichkeiten zur Textgestaltung benötigt.

Der britischen Firma Tasman-Software muß man bescheinigen, daß sie den richtigen »Riecher« besitzt. Die erste Version von Tasword lief noch auf dem Sinclair ZX81 (Stichwort: Folientastatur). Erst später wurde das Programm an andere Computer angepaßt, die mit dem 8086/88 von Intel oder der Z80 von Zilog arbeiten. Sinclair-Spectrum, Memotech MTX, Schneider CPC, Joyce und neuerdings sogar IBM-PC lautet die Reihenfolge. Tasword ist damit das einzige Textverarbeitungsprogramm, das sowohl auf dem ZX81 als auch auf dem IBM-PC läuft!

Neben der englischen Originalversion für den CPC (69,90 Mark) ist das deutsche Tasword-D (99,90 Mark) und Tasword-6128 (99 Mark) erhältlich. Tasword-6128 läuft ausschließlich auf dem CPC-6128 und ist eines der Programme, die auf den erweiterten Speicher von 128 KByte unter dem Betriebssystem Amsdos zugreifen.

Das Aussehen und die Leistungsfähigkeit von Tasword hat sich von Version zu Version geändert, doch das grundsätzliche Konzept ist gleichgeblieben. Tasword ist im Editiermodus hauptsächlich tastengesteuert. Hier ähnelt es in der Arbeitsweise Wordstar. Die Tasten sind jedoch anders belegt. So formatieren Sie zum Beispiel mit <CTRL+J> einen Absatz neu, während <CTRL+K> eine einzelne Zeile umbricht.

Im Gegensatz zu Wordstar werden die Sondertasten des Computers auch ohne gesonderte Anpassung in die Bedienung integriert. Mit den vier Pfeiltasten bewegen Sie den Cursor. Kombiniert man die Cursortasten mit der <Shift>-Taste, so springt der Cursor eine Bildschirmseite vor- oder rückwärts, beziehungsweise an den Zeilenanfang oder das Zeilenende. <CTRL+↑> bewegt den Cursor an den Textanfang, und <CTRL+↓> an das Ende des Textes.

## Komfortabel mit Hilfsmenüs

Ähnlich wie bei Wordstar, ist in der oberen Bildschirmhälfte ein Hilfsmenü eingeblendet. Es zeigt die wichtigsten Tastenkombinationen für Formatieren, Löschen, Einfügen und Steuerung des Cursors an. Da sich durch das große Menü die Arbeitsfläche auf dem Bildschirm stark verkleinert, empfiehlt es sich, möglichst bald nach der Eingewöhnung das Menü auszublenden, um das Textfenster zu vergrößern. Wenn Sie in Schwierigkeiten geraten, können Sie das Hilfsmenü jederzeit wieder einschalten. Außerdem erscheint nach Drücken von <ESC> ein großes Menü, das den gesamten Bildschirm einnimmt und durch <ENTER> wieder verschwindet.

Insgesamt lassen sich fünf verschiedene Hilfsmenüs zusammen mit dem Text darstellen. Dabei ist sehr praktisch, daß der Benutzer mit den Tasten <CTRL+↑> und <CTRL+↓> in den Menütexten nach oben und unten scrollen kann. So steht immer der gerade benötigte Hilfstext auf dem Bildschirm. Die weiteren Funktionen von Tasword entsprechen weitgehend dem Standard, der auch von anderen Textverarbeitungsprogrammen eingehalten wird.

Jede Textzeile kann unter Tasword bis zu 128 Zeichen umfassen. Da der CPC jedoch maximal 80 Zeichen pro Bildschirmzeile darstellt, scrollt Tasword den Bildschirm nach links oder rechts. Dies geht – besonders im Vergleich zu Wordstar – erstaunlich schnell und läßt die Tatsache vergessen, daß ein großer

Teil von Tasword in Basic geschrieben wurde.

Der Druck auf die <TAB>-Taste führt dazu, daß der Computer den nächsten Tabulatorstop anspringt. <SHIFT+TAB> erlaubt das Setzen und <CTRL+TAB> das Löschen von Tabulatormarken. <CTRL+X> setzt die Tabulatoren auf die Standard-Einstellung zurück.

## Tabulatoren total

Die jeweils ausgewählten Tabulatorstops erscheinen am unteren Rand des Bildschirms. In dieser Statuszeile können Sie auch den linken und rechten Rand ablesen sowie erkennen, ob der Randausgleich, der Wortumbruch, der Einfüge-Modus und die Darstellung des Seitenumbruchs ein- oder ausgeschaltet sind. Zusätzlich wird die aktuelle Position des Cursors in Zeile und Spalte angegeben.

Tasword kann Kopf- und Fußzeilen verwalten. Kontrollzeichen für den Drucker lassen sich vor dem Ausdrucken eingeben. Hier gewinnt Tasword ganz eindeutig gegenüber Wordstar. Während Wordstar universell an alle nur erdenklichen Druckertypen anpaßbar ist, und deshalb nur die notdürftigsten Funktionen standardmäßig bereithält, arbeitet Tasword bewußt nur mit Epson-kompatiblen Druckern.

Dazu gehören auch Schneider NLQ-401, der baugleiche Brother M-1009 und das Nachfolgemodell Schneider DMP-2000. Hervorhebungen, Unterstreichungen, Kursivschrift, Hoch- und Tiefstellung von Zeichen, Schmalschrift, Doppeldruck, Zeilenabstand und eine Vielzahl weiterer Druckerfunktionen werden von Tasword unterstützt.

Ein besonderes Plus ist die Fähigkeit, Grafiksymbole aus dem erweiterten Zeichensatz des CPC in den Text einzubinden. So lassen sich auch diejenigen Zeichen, deren ASCII-Code zwischen 128 und 255 liegt, auf Drucker ausgeben.

Tasword besitzt eine Software-Schnittstelle zum Programm Tasprint, das ebenfalls von Tasman-Software geschrieben wurde. Tasprint gibt Schriftzeichen aus verschiedenen Schriftarten als hochauflösende Grafik auf den Drucker aus. Die Herstellung von Briefen und Einladungen in grafisch aufwendigen Schriftarten (zum Beispiel Fraktur) ist damit eine leichte Übung.

Ist ein Text fertig bearbeitet, wird der Texteditor verlassen, und zwar mit <CTRL+ENTER>. Tasword zeigt dann ein Menü an, das Optionen zum Drucken des Textes, zum Zugriff auf die Diskette und zur Beendigung des Programms anbietet.

Will man den Text auf den Drucker ausgeben, muß man eine Reihe von Fragen zur Formatierung beantworten. Dazu zählt die Angabe des gewünschten Textausschnitts, des Zeilenabstands und der Anzahl der zu druckenden Kopien. Sie müssen sich entscheiden, ob Sie Endlospapier oder einzelne Blätter verwenden wollen, ob Kopf- und Fußzeilen nötig sind und ob die Seiten durchnumeriert werden sollen.

Um die vorgegebenen Werte der Reihe nach zu übernehmen, reicht es aus, die <ENTER>-Taste zu betätigen. Man kann aber auch alle voreingestellten Angaben auf einmal übernehmen, indem man die <COPY>-Taste drückt.

Alles in allem ist Tasword ein Programm, das eine Vielzahl von Befehlen und Funktionen zur Textbearbeitung besitzt. Weil der Textbuffer von 13 KByte in der 64-KByte-Version des CPC recht klein ist, lassen sich jedoch keine längeren Texte schreiben. (Diese Einschränkung gilt natürlich nicht für Tasword-6128.)

Aufgrund des niedrigen Preises ist Tasword für alle Anwender ausreichend, die primär ihre private Korrespondenz mit dem Computer erledigen. Für Geschäftsanwendungen ist Tasword jedoch eine Nummer zu klein.

## Star oder Sternchen?

Das Programm »Star-Texter« aus dem Sybex-Verlag versteht sich nicht ausschließlich als Textverarbeitungssystem, sondern erhebt bei einem Preis von 85 Mark auch den Anspruch, ein Trainingsprogramm zu sein.

Star-Texter bietet eine neue Form der Tastensteuerung an. Es wird nicht wie Wordstar und Tasword ausschließlich über umständliche Control-Kombinationen bedient (wenngleich sie vorhanden sind), sondern ist in seiner Bedienung an das Textverarbeitungsprogramm Word angelehnt. Aus diesem Grund sind keine umfangreichen Hilfsbildschirme notwendig. Statt dessen ist die wichtigste Taste bei Star-Texter <ESC>. Sobald diese Taste gedrückt wird, erscheint in der oberen Hälfte des Bildschirms das Hauptmenü, das eine Bildschirmzeile umfaßt und die Funktionen »Format«, »Block«, »Suchen«, »Grafik«, »Drucken«, »Archiv« und »Parameter« auflistet.

Die einzelnen Menüpunkte wählen Sie durch Angabe des Anfangsbuchstabens. Dabei kann durchaus ein weiteres Menü erscheinen. Dieses Untermenü ist vom Prinzip her genauso aufgebaut und stellt Unterfunktionen zum oberen Menüpunkt zu Verfügung.

In der Editor-Betriebsart erfolgt die Cursorsteuerung über die Pfeiltasten. Im Gegensatz zu Wordstar läßt sich der Cursor über den gesamten Bildschirm bewegen, auch auf diejenigen Stellen, die noch nicht beschrieben sind – folglich ein vollwertiger Full-Screen-Editor. In Kombination mit der <SHIFT>-Taste bewegen die Pfeiltasten den Cursor an Anfang oder Ende des Textes beziehungsweise der Zeile.

Die <CTRL>-Taste hat in Verbindung mit den Pfeiltasten eine ungewöhliche Funktion. Mit <CTRL+↑>) wird eine Zeile gelöscht, und mit <CTRL+↓> eine Leerzeile eingefügt.

Angenehm fällt die hohe Geschwindigkeit auf, mit der Star-Texter die Cursorbewegungen durchführt. Da macht sich bemerkbar, daß Star-Texter vollständig in Maschinensprache geschrieben wurde.

## Öfter mal was Neues

Die obere Bildschirmzeile zeigt ein Zeilenlineal, das den rechten Rand und die vorgewählten Tabulatorpositionen markiert. Etwas ungewöhnlich ist die Tatsache, daß die Einstellung des rechten Randes fest ist, die des linken Randes jedoch nur bis zum nächsten Absatz gilt.

Positiv anzumerken ist eine Markierung im Zeilenlineal, die auf Höhe des Textcursors mitläuft. Dies erleichtert das Ablesen der aktuellen Spaltenposition. Einen mitlaufenden Cursor ist man sonst nur von speziellen Textcomputern her gewöhnt. Die Zeilenposition des Cursor wird jedoch inkonsequenterweise als Zahlenwert am unteren Bildschirmrand angezeigt.

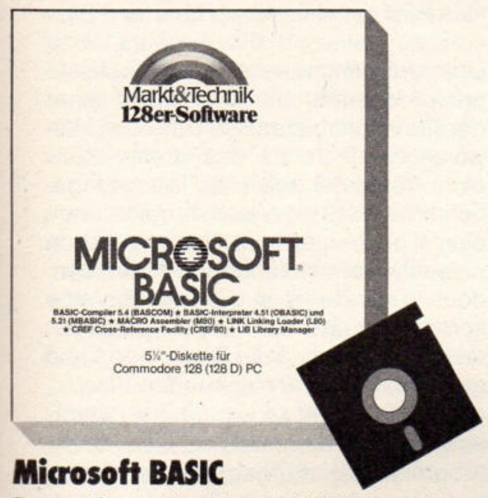
Ansonsten offeriert Star-Texter die üblichen Befehle zur Textverarbeitung, wie Blocksatz, Zeilenumbruch, Trennen von Wörtern sowie Suchen und Ersetzen von Zeichenfolgen. Daß man auch bei der Suche nach einer Zeichenfolge eine Ersatz-Zeichenfolge angeben muß, macht sich störend bemerkbar.

Das Druckmenü bietet dem Benutzer die Auswahl zwischen »Drucken«, »Layout« und »Daten mischen«. Unter dem letztgenannten Punkt ist eine Serienbrief-Option integriert, die Sie allerdings nur in Verbindung mit dem Programm Star-Datei (ebenfalls von Sybex) nutzen können.

Die Funktion »Layout« ist bei Textverarbeitungsprogrammen für CPCs eine lobenswerte Neuheit. Normalerweise ist es recht schwierig herauszufinden, wie ein Text nach der Ausgabe auf den Drucker wirklich aussieht. Dazu kann der CPC zu wenige Zeilen auf dem Bildschirm darstellen. Ein Probeausdruck ist in den meisten Fällen unumgänglich.




# Leistungsfähige Programmiersprachen für Commodore 128 und Schneider-Computer










## Microsoft BASIC

Das umfassende Microsoft-BASIC- und Assembler-Entwicklungspaket enthält:

- BASIC-Compiler 5.4 (BASCOM)
- BASIC-Interpreter 4.51 (OBASIC) und 5.21 (MBASIC)
- MACRO Assembler (M80)
- LINK Linking Loader (L80)
- CREF Cross-Reference Facility (CUEF 80)
- LIB Library Manager

für den effizienten Einsatz kaufmännischer und technisch-wissenschaftlicher Anwendungen.

**Hardware-Anforderungen für Commodore 128/128D:**
Diskettenlaufwerk, Betriebssystem CP/M 3.

**Hardware-Anforderungen für Schneider-Computer:**
CPC 464, 664, 6128 oder Joyce, ein Diskettenlaufwerk, Betriebssystem CP/M 2.2 oder CP/M Plus. Der Interpreter erfordert mindestens 32 K Speicher, der Compiler und der Makroassembler mindestens 48 K.

| | Version | Best.-Nr. | Format | Preis DM | sFr | öS |
|---|---|---|---|---|---|---|
| Microsoft BASIC | Schneider CPC, Joyce | MS 617 | 3" | 199,-* | 178,- | 1990,-* |
| Microsoft BASIC | Commodore 128/128D | MS 627 | 5 1/4" | 199,-* | 178,- | 1990,-* |
| Pascal/MT+ | Schneider CPC, Joyce | MS 611 | 3" | 174,-* | 158,- | 1690,-* |
| Pascal/M+ | Commodore 128/128D | MS 621 | 5 1/4" | 174,-* | 158,- | 1690,-* |
| CBASIC Compiler | Schneider CPC, Joyce | MS 612 | 3" | 174,-* | 158,- | 1690,- |
| CBASIC Compiler | Commodore 128/128D | MS 622 | 5 1/4" | 174,-* | 158,- | 1690,-* |

* inkl. MwSt. Unverbindliche Preisempfehlung

## Pascal/MT+

Pascal/MT+ ist ein volles ISO-Standard-Pascal, das um eine leistungsfähige Programmierumgebung für Industrie-, Geschäfts- und Ausbildungseinsatz sowie Möglichkeiten zur Systemprogrammierung erweitert wurde.

**Hardware-Anforderungen für Commodore 128/128D:**
ein Diskettenlaufwerk, Betriebssystem CP/M 3.

**Hardware-Anforderungen für Schneider-Computer:**
CPC 464 und CPC 664 (mit Speichererweiterung), dem CPC 6128 und dem PCW 8256 (Joyce) unter CP/M und CP/M Plus. Kompilierte Programme sind bei entsprechender Größe, auch auf dem CPC 464 und CPC 664 ohne Speichererweiterung lauffähig.

## CBASIC-Compiler

Der Hochleistungs-BASIC-Compiler für Softwareprofis zur Erstellung kommerzieller Anwendungen.
Der CBASIC-Compiler ist ein Compiler, der Maschinencode erzeugt und die Programmierung und den Test separater Module erlaubt, die später ein komplettes Programm ergeben sollen. Die integrierten Grafikmöglichkeiten des CBASIC-Compilers erlauben die Programmierung vielseitiger Grafikprogramme für eine Vielzahl von Anwendungen (nur auf Computern mit GSX-Software).

**Hardware-Anforderungen für Commodore 128 PC:**
ein Diskettenlaufwerk, Betriebssystem CP/M 3.

**Hardware-Anforderungen für Schneider-Computer:**
Der CBASIC-Compiler läuft auf Schneider CPC 464 mit Diskettenlaufwerk DDI-1, dem CPC 664, dem CPC 6128 und dem PCW 8256 (Joyce). Für Grafikprogramme wird die GSX-Software benötigt, die nur mit dem CPC 6128 und PCW 8256 (Joyce) ausgeliefert wird.

Diese Markt & Technik-Software erhalten Sie in den Fachabteilungen der Warenhäuser, bei Ihrem Computerfachhändler, im Buchhandel oder direkt beim Verlag gegen Vorauskasse.
Fragen Sie auch nach dem neuen Gesamtverzeichnis Herbst '86, oder fordern Sie es direkt beim Verlag an.




Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Tel. (0222) 677526 · Ueberreuter Media Verlagsges. mbH, Alser Straße 24, A-1091 Wien, Tel. (0222) 481538-0.

608202

»Layout« zeigt hingegen ein symbolisches Blatt Papier auf dem Bildschirm. Darauf wird jeder Buchstabe durch einen Grafikpunkt repräsentiert. So wird sehr gut der Eindruck vermittelt, wie der Text auf dem Papier aussehen wird.

Das Verhalten von Star-Texter in bestimmten Situationen und die Standard-Einstellungen legt der Benutzer im Menü über »Parameter« fest. Hier werden Werte, wie Zahl der Leerzeichen bei Einrückungen, gewünschter Zeichensatz, Zahl der Zeilen pro Seite, Zeilenabstand auf dem Drucker sowie Abstand der Kopf- und Fußzeilen vom übrigen Text eingestellt. Auch die Bildschirmfarben lassen sich den eigenen Wünschen anpassen.

Die einzelnen Werte werden im Parameter-Teil nicht durch Eingabe von Zahlenwert eingetragen. Statt dessen drücken Sie die rechte oder linke Pfeiltaste, je nachdem, ob Sie den Wert vermindern oder erhöhen wollen. Dabei ist für alle Eingaben ein Wertebereich von 0 bis 255 vorgesehen – auch in solchen Fällen, in denen es nicht sinnvoll ist. So ist zum Beispiel die Farbe 64 genauso schwarz wie die Farbe 0. Und was bei der Frage nach Einzelblatt oder Endlospapier statt »ja« und »nein« ein Zahlenwert zwischen 0 und 255 bedeuten soll, mag Geheimnis der Programmautoren bleiben.

Dafür lassen sich die Parameter dauerhaft auf Diskette speichern, so daß der Anwender sie nicht nach jedem Programmstart neu eingeben muß. Selbst Wordstar bietet diese Fähigkeit nicht.

Nur wenige Textverarbeitungsprogramme können mehrspaltigen Text erzeugen. Der Text, den Sie gerade lesen, ist zum Beispiel dreispaltig. Star-Texter kann immerhin zweispaltigen Text bearbeiten. Dieser wird sogar auf dem Bildschirm angezeigt. Das Eingeben beziehungsweise Formatieren von zweispaltigen Texten mit Star-Texter ist jedoch ziemlich aufwendig.

## Grafik im Text

Star-Texter erlaubt es, Grafiken in Texte einzubinden. Allerdings dürfen Sie kein komfortables Zeichen-Unterprogramm erwarten, mit dem der Anwender aufwendige Bilder zeichnen und in den Text einfügen kann. Vielmehr müssen Sie jedes Grafikzeichen einzeln als Blockgrafik definieren und anschließend die Zeichen zum gewünschten Bild zusammensetzen.

Da Star-Texter umfangreiche Funktionen zur Druckersteuerung integriert hat, müssen Sie das Programm an Ihren Drucker anpassen. Dazu befindet sich auf der gelieferten Diskette das (ungeschützte) Basic-Programm »drucker. bas«. Es hat eine ansprechende Bildschirmgestaltung und gestattet dem Benutzer, für jede Druckerfunktion eine bis zu acht Byte lange Escape-Sequenz anzugeben.

Bei der Gesamtbeurteilung von Star-Texter ist das Urteil etwas gemischt. Auf der einen Seite heben sich die hohe Arbeitsgeschwindigkeit, die Fähigkeit zur Darstellung von Grafiken und zweispaltigem Text sowie andere kleine Annehmlichkeiten hervor. Auf der anderen Seite muß jedoch die teilweise inkonsequente Ausführung des Programms (Anzeige der Cursorposition in der Zeile, Suchen mit Ersatzwort, Parameter-Menü) bemängelt werden. Zudem sind verschiedene Funktionen so gut auf den Control-Tasten versteckt, daß man sie nur mit Mühe wiederfindet. Vielleicht fehlte hier etwas Zeit bei der Programmentwicklung? Denn daß der gute Wille da war, beweist die Programmgestaltung eindeutig.

## ... und noch ein Sternchen

Das Programm Star-Writer der Firma Star-Division ist mit 198 Mark – ebenso wie Wordstar – in der gehobenen Preisklasse angesiedelt, was eine gewisse Erwartungshaltung beim Testen hervorruft. Um es gleich vorwegzunehmen, die Erwartungen erfüllten sich keineswegs.

Star-Writer ist in Turbo-Pascal geschrieben. Das ist an sich kein Nachteil, da das Programm jedoch sehr umfangreich ist, müssen ständig Overlay-Dateien von der Diskette nachgeladen werden. Das kostet Zeit und zwingt bei Benutzung eines einzelnen Laufwerks zu ständigem Diskettenwechsel.

Star-Writer geht bei der Benutzerführung einen anderen Weg, als die bisher besprochenen Programme. Nach dem Start des Programms erscheinen auf dem Bildschirm mehrere große Piktogramme, die die einzelnen Programmteile symbolisieren.

Eine Schreibmaschine repräsentiert den Texteditor, ein Bleistift den integrierten Grafikeditor und ein Karteikasten die Adressenverwaltung. Der Zeicheneditor wird durch einige umdefinierte Buchstaben dargestellt, und die DFÜ-Routine zur Datenfernübertragung durch einen Akustikkoppler, der an einen schematisierten Rechner angeschlossen ist.

Mit den Pfeiltasten steuern Sie nun eine Linie, die vom Mittelpunkt des Bildschirms ausgeht, zum jeweils gewünschten Programmteil. Die Wahl wird durch Drücken der kleinen ENTER-Taste bestätigt. Das Verfahren ist zwar etwas verspielt, zeigt sich aber als recht zweckmäßig.

Nach Aufruf des Programmteils »Textverarbeitung« wird dieser von Diskette nachgeladen. Er listet das Inhaltsverzeichnis der eingelegten Datendiskette auf. Die obere Bildschirmzeile zeigt ständig den Programmnamen an – eine unnötige Platzverschwendung. Ein Zeilenlineal an dieser Stelle wäre wesentlich sinnvoller gewesen.

In der zweiten Bildschirmzeile stellt der Computer eine Menüleiste dar, die entfernt an die Leisten von GEM auf Atari ST und Schneider PC erinnert. Allerdings wird hier mangels Hardware kein Pfeil mit einer Maus über den Bildschirm gesteuert. Die Auswahl eines Unterprogramms aus der Menüleiste erfolgt vielmehr durch Drücken einer der Funktionstasten. So ruft beispielsweise die Taste F1 das »Bearbeiten« einer Textdatei auf. Hier fällt unangenehm auf, daß man jede Angabe noch einmal bestätigen muß. Dadurch lassen sich zwar Fehlbedienungen vermeiden, doch in der Regel ist die erste Eingabe korrekt, so daß das ewige Bestätigen den Anwender auf die Dauer nervt und zu ärgerlichen Verzögerungen führt.

Und dann geht es wieder ans Diskettenwechseln. Mehrmals müssen Sie die Programmdiskette gegen Datendiskette austauschen, bevor Sie in die eigentliche Textverarbeitung einsteigen.

Sobald man mit dem Schreiben beginnt, zeigt sich ein weiteres Ärgernis: Bei schneller Eingabe von Buchstaben »hängt« die Bildschirmanzeige oft bis zu zwanzig Zeichen nach. Hier macht sich deutlich bemerkbar, daß das Programm nicht in Maschinensprache, sondern in einer Hochsprache geschrieben wurde. Gemächlicheren Schreibern fällt dies freilich kaum auf.

Wenn Sie mit der Funktionstaste 1 den Menüpunkt »Layout« aufrufen, finden Sie die Einstellungen für linken und rechten Rand, Blocksatz, Zahl der Zeilen pro Blatt und Abstand der Kopf- und Fußzeilen vom übrigen Text. Seltsam ist, daß die untere Bildschirmzeile beim Bewegen des Menübalkens in diesem Pull-down-Menü entsetzlich flimmert, während sie dies bei allen anderen Pull-down-Menüs nicht macht.

Der Befehlsumfang von Star-Writer ist recht groß. Alle wichtigen Funktionen wie Festlegen von Tabulatoren und Rändern, Blocksatz, Trennhilfen, Rechnen im Text, Kopieren und Löschen von Textblöcken sowie Suchen und Ersetzen sind vorhanden. Da Star-Writer jedoch ausschließlich mit Pull-down-Menüs arbeitet, geht die Arbeit an einem Text für geübte Schreiber erheblich langsamer vonstatten als bei einem

herkömmlichen Textverarbeitungsprogramm mit Tastatursteuerung wie Wordstar, Tasword oder Star-Texter.

Star-Writer ist das einzige getestete Programm, das seitenorientiert arbeitet. Seitenorientiert heißt, sobald Sie am Ende einer Textseite angekommen sind, wird der bisher eingegebene Text gespeichert. Das erfordert wieder einen mehrmaligen Diskettenwechsel. Wenn dieser Vorgang abgeschlossen ist, können Sie die nächste Textseite bearbeiten.

Für Textänderungen am Ende einer Seite ist das Konzept der seitenorientierten Verarbeitung schlichtweg als katastrophal zu bezeichnen. Sobald Sie mit dem Cursor auf die nächste Seite kommen, wird der alte Text gespeichert und die nächste Textseite von der Diskette geladen. Bemerken Sie jetzt erst einen Fehler und wollen mit dem Cursor auf die alte Seite zurückgehen, muß die Seite erst wieder zeitaufwendig geladen werden. Welch ein modernes Textverarbeitungsprogramm arbeitet so umständlich?

Die anderen mitgelieferten Hilfsprogramme, die Sie ebenfalls im Hauptmenü von Star-Writer auswählen, sind brauchbar, aber keineswegs überragend. Die Funktion »Grafik« stellt einen Briefkopf-Editor bereit, »Zeichen« einen Symbol-Editor, und »Adreßverwaltung« sowie »DFÜ« sprechen für sich.

Kleinere Schönheitsfehler, die in einer neuen Version des Programms korrigiert werden sollten, betreffen die Rechtschreibung in einzelnen Programmteilen (»Star-Writer«, »Starwriter«, »Star Writer«) und die Inkonsequenz, daß manchmal das Drücken einer beliebigen Taste akzeptiert wird, dann aber wieder nur die kleine Enter-Taste erlaubt ist.

Der Käufer bekommt mit der Programmdiskette nicht ein einzelnes Handbuch, sondern deren zwei! Dabei handelt es sich um das Handbuch für die Version 3.0 sowie das Manual zur vorigen Programmversion. Das Vorwort des neuen Handbuchs begründet die doppelte Lieferung lapidar damit, daß das neue Handbuch nicht so ausführlich wie das alte sei. Wenn beim Benutzer noch Fragen offen sind, soll er doch im alten Handbuch nachschlagen!

## Kein Kopieren bei Textomat

Zusammenfassend gesagt ist es positiv zu werten, daß ein Software-Hersteller versucht, die Ergebnisse moderner Software-Ergonomie (zum Beispiel Pull-down-Menüs) in sein Textprogramm zu integrieren. Da jedoch Mängel wie langsame Arbeitsgeschwindigkeit und seitenorientierte Textverarbeitung überwiegen, fällt es schwer, für dieses Programm eine Empfehlung auszusprechen.

»Textomat« von Data-Becker (99 Mark) ist das einzige Programm im Test, das mit Kopierschutz vertrieben wird. Das ist bedauerlich, weil dem ehrlichen Käufer die Möglichkeit genommen wird, Sicherheitskopien der wertvollen Originaldiskette anzulegen. Für einen Betrag von 20 Mark kann man allerdings von Data-Becker eine Sicherheitskopie – wiederum mit Kopierschutz – anfordern.

Nach dem Programmstart fordert Textomat die Eingabe mehrerer Parameter. Der Benutzer muß die gewünschten Bildschirmfarben, deutsche oder amerikanische Tastaturbelegung und den Namen für die Druckerparameter-Datei wählen. Die Eingabe des aktuellen Datums ist nicht zwingend.

Danach gelangt der Anwender in den Schreib-Modus. Im Gegensatz zu den anderen getesteten Programmen belegt Textomat die noch nicht beschriebenen Bildschirmpositionen mit Punkten. Was man nun bevorzugt, ist reine Geschmackssache.

Textomat wurde auf Geschwindigkeit getrimmt. Die Zeichenausgabe ist so flink, daß selbst Schnellst-Schreiber keine Chance haben, die Eingaberoutine zu »überholen«. Besonders im Vergleich zu Star-Writer fällt die hohe Verarbeitungsgeschwindigkeit positiv auf. Immerhin handelt es sich bei Textomat um eine 24 KByte große Binärdatei, also reinen Maschinencode.

Doch die Programmautoren haben es in ihrem Bemühen um hohe Arbeitsgeschwindigkeit teilweise auch übertrieben. So wurden neue Scroll-Routinen programmiert, die Textomat anstelle der Routinen des Betriebssystems verwen-

# Die wichtigsten Textverarbeitungsprogramme im Überblick

| Programmname | 464 | 664 | 6128 | CP/M erforderlich | bei 464/664 Speichererweiterung erforderlich | Kassette | 3-Zoll-Diskette | 5¼-Zoll-Diskette | Serienbriefoption | Preis in Mark | Anbieter | Besonderheiten |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Easy Topword | x | | | | | x | | | | 79,– | Schneider | – |
| Protext | x | x | x | | | x | x | | x | 79,– | Denisoft | auf Diskette 99,– Mark, als ROM-Modul 158,– Mark |
| Star-Texter | x | x | x | | | | x | | x | 85,– | Sybex | Zeichensatz frei definierbar, Layout-Funktion |
| Star-Writer | x | x | x | x | x | | x | x | x | 198,– | Star Division | Adreßverwaltung, DFÜ-Funktion |
| Tasword | x | | | | | x | | | x | 69,90 | Profisoft | – |
| Tasword-D | x | | | | | | x | | x | 99,90 | Profisoft | Zeichensatz frei definierbar, deutsche Version |
| Tasword 6128 | | | x | | | | x | | x | 99,– | Profisoft | Zeichensatz frei definierbar |
| Tex Pack | x | x | x | x | | | x | | x | 198,– | Schneider | mit Adreßverwaltung |
| Textomat | x | x | x | | | | x | | x | 99,– | Data Becker | vollständig menügesteuert |
| Textverarbeitung | x | x | x | | | x | x | | x | 59,– | Data Media | auf Diskette 69,– Mark |
| Wordstar 3.0 | x | x | x | x | | | x | x | x | 199,– | Markt& Technik | – |
| Workwriter Junior | | | x | x | | | x | | x | 145,– | Gepo-Soft | IBM-Zeichensatz wird unterstützt |

det. Die neuen Routinen sind zwar ausgesprochen schnell, doch der Bildschirm beginnt zu flimmern. Da der Benutzer aber auch auf langsames Scrollen umschalten kann, entsteht dadurch kein Nachteil.

Textomat arbeitet menügesteuert. In der untersten Bildschirmzeile zeigt es die Grundfunktionen »Edit«, »Formular«, »Ausgabe« und »Dienst« an. Während der Texteingabe steht in der obersten Bildschirmzeile die Meldung »Schreib-Modus«. Mit <CTRL+ENTER> gelangt man ins Menü. Hier bewegen Sie mit den Pfeiltasten einen inversen Balken über die angegebenen Menüpunkte. Eine Betätigung der ENTER-Taste führt die Funktion aus.

Mit »Edit« rufen Sie ein neues Menü für das Einlesen von Dateien, Suchen, Löschen von Teilen des Textes und die Blockoperationen »Mark«, »Schieb«, »Kopier«, »Lösch« und »Speicher« auf.

»Formular« ist eine Besonderheit von Textomat. Für jeden Text läßt sich ein Formular anlegen, das die Eigenschaften des Textes beschreibt. Rechten und linken Rand, Blattlänge, Zeichendichte, Block- oder Flattersatz, Einzelblattbetrieb und Proportionalschrift legt der Anwender in dieser Funktion fest.

## Praktische Formular-Funktion

Über »Ausgabe« rufen Sie die Menüpunkte »Zeigen«, »Speichern«, »Drukken« und »Rundschreiben« auf. Textomat zählt nicht zu den Programmen, die den Text auf dem Bildschirm bereits so darstellt wie er auf dem Drucker ausgegeben wird, also mit Blocksatz, Trennungen etc. Statt dessen fügt der Benutzer Steuerzeichen in den Text ein, die das Programm beim Ausdrucken an der entsprechenden Stelle zum Drukker übergibt. Um Probedrucke überflüssig zu machen, führt Ihnen die Funktion »Zeigen« den Text in etwa so vor, wie ihn später der Drucker ausgibt.

Textomat ist unter den besprochenen Programmen das einzige, das neben der 80-Zeichen-Darstellung auch im 40-Zeichen-Modus arbeitet. Dieser macht zwar die Eingabe des Textes unübersichtlicher, ist aber allen Anwendern zu empfehlen, die Schwierigkeiten haben, auf dem Farbmonitor den Text zu lesen.

Insgesamt gesehen ist Textomat ein leistungsfähiges Programm. Im Gegensatz zu Programmen wie Star-Texter ist es konsequent menügesteuert. Wer die Menüsteuerung der Tastensteuerung vorzieht und auf hohe Geschwindigkeit Wert legt, dem kommt Textomat gerade recht. (Martin Kotulla/ma)

# Pascal nach Wahl

**Schon vor langer Zeit führte die englische Firma Hisoft den Pascal-Compiler Pascal80 speziell für CP/M-Computer ein. Jetzt bietet die neue Version Pascal 80V2 Turbo-Pascal, dem Star unter den Pascal-Compilern, Paroli.**

Turbo-Pascal ist seit Jahren unangefochtener Spitzenreiter unter den Pascal-Compilern für die Betriebssysteme CP/M und MS-DOS. Neben dem günstigen Preis zeichnen die überragenden Leistungsmerkmale, wie hohe Geschwindigkeit und komfortable Bedienung, für den Erfolg verantwortlich. Was mag da eine – im Vergleich zu Borland – winzige Software-Firma wie Hisoft dazu veranlaßt haben, einen eigenen Pascal-Compiler für CP/M auf den Markt zu bringen? Lohnt sich das Angebot für den engagierten Programmierer?

Klassische (man könnte auch sagen: »altmodische«) Compiler bestehen aus mehreren getrennten Programmen. Der Programmierer gibt zuerst mit dem Texteditor das Programm im Quellcode ein. Im nächsten Schritt ruft er den Compiler auf, der einen Zwischencode produziert. Aus diesem Zwischencode erzeugt dann der Linker das lauffähige Programm. Das alles läuft natürlich sehr umständlich und langsam ab. Besonders, wenn der Compiler im Quellcode einen Fehler findet, entwickelt sich die Weiterarbeit recht aufwendig. Dann nämlich muß der Editor wieder geladen, der Fehler ausgebessert, die Datei gespeichert und der Compiler neu gestartet werden.

Hisoft-Pascal80 funktioniert nach einem ähnlichen Prinzip. Lediglich der Linker (Programmbinder) wird gespart. Der Compiler selbst erzeugt die lauffähigen Dateien.

Auf der Diskette von Hisoft befindet sich auf Seite A die CP/M 2.2-Version von Pascal80, die speziell auf den Schneider CPC 464 und CPC 664 zugeschnitten ist. Auf Seite B findet der Besitzer eines CPC 6128 oder eines Joyce die Anpassung an CP/M Plus.

## Pascal 80 gegen Turbo

Der Editor heißt ED80.COM und ist schon von anderen Compilern von Hisoft bekannt. Er ist – ebenso wie der Turbo-Pascal-Editor – befehlskompatibel zum Textverarbeitungssystem Wordstar. Natürlich fallen einige zum Editieren unnötige Funktionen unter den Tisch. Dazu gehören Routinen zum Umformatieren von Absätzen, Wortumbruch und Trennhilfen. ED80 arbeitet sehr schnell. Im Gegensatz zum Turbo-Editor ist allerdings der Bildschirmaufbau unruhiger. So schaltet der Hisoft-Editor beim Löschen von Textzeilen mit <CTRL+Y> kurzzeitig die Bildausgabe ab und baut dann den gesamten Bildspeicher neu auf. Nützlich erweist sich die Anzeige des jeweils letzten Befehls und des noch verfügbaren Speicherplatzes.

Der Turbo-Pascal-Editor und ED80 stammen zwar beide von Wordstar ab, wurden aber teilweise in verschiedene Richtungen weiterentwickelt. So beendet ED80 seine Arbeit nur mit <CTRL +K> und <X>, nicht aber mit <CTRL+K> und <D>. Beim Editor von Borland ist das genau umgekehrt.

Das automatische Einrücken von Programmzeilen, wie es in Pascal wichtig ist, wird mit <CTRL+O> und <I> eingeschaltet. <CTRL+J> ruft einen Hilfstext auf, der alle Tastenkombinationen listet. Eine sehr wertvolle Erweiterung gegenüber den üblichen Funktionen von Wordstar ist das Auffinden einer Programmzeile einfach durch Angabe ihrer Zeilennummer.

Mit einigem Erstaunen stellt man fest, daß Pascal80 mit zwei Compilern geliefert wird: HP.COM und HP80.COM. HP80 ist die ältere Version vom 30. Mai 1986. HP ist die erweiterte und drei KByte längere Implementation vom 19. September 1986. Der »alte« Compiler dient nur der Kompatibilität mit älteren Pascal80-Quellcodes.

Compiler der Firma Hisoft verarbeiten meist nicht den vollständigen Sprachumfang. So fehlen bei Hisoft-C beispielsweise die Fließkommazahlen und Bit-Felder. Hisoft-Pascal unter Amsdos kennt die Datentypen FILE und TEXT nicht. Ebenso sind dort variante Re-

cords, Prozeduren und Funktionen als Parameter unzulässig.

Die neuere der beiden CP/M-Versionen erlaubt variante Records und den Zugriff sowohl auf Text- als auch auf Binärdateien. Abgesehen davon, daß immer noch keine prozeduralen und funktionalen Parameter compiliert werden, erfüllt Pascal80 jetzt den Standard von Niklaus Wirth – sogar besser als Turbo-Pascal, denn die Standard-Prozeduren PUT und GET unterstützt Turbo-Pascal nicht. Was bei einem Compiler besonders interessiert, ist die Effektivität beim Übersetzen. Darunter versteht man die Größe und Geschwindigkeit der erzeugten Datei. Als Beispiel dient uns das Demonstrationsprogramm »Primes« von der Pascal80-Diskette. Es läuft mit einer minimalen Änderung (die Compiler-Direktive muß aus dem Quelltext entfernt werden) auch unter Turbo-Pascal und eignet sich damit sehr gut als Vergleichsobjekt. Primes berechnet alle Primzahlen zwischen 1 und 20499. Das Programm arbeitet mit dem Algorithmus »Sieb des Eratosthenes«.

Pascal80 erzeugt einen Objektcode von 6 KByte Länge, Turbo-Pascal hingegen ein 9 KByte langes Programm. Diese Runde gewinnt also eindeutig Pascal80. Bei der Laufzeit des erzeugten Codes hingegen muß es sich geschlagen geben. Während der Code von Turbo-Pascal in 1,58 Minuten alle Primzahlen (einschließlich Ladezeit) findet, benötigt das gleichwertige Pascal 80-Programm immerhin 2,44 Minuten. Der Name »Turbo« kommt also nicht von ungefähr. Allerdings sollte man solche Benchmark-Tests nicht überbewerten. Denn die meiste Zeit verharren Programme immer noch beim Warten auf Benutzereingaben oder bei der Kommunikation mit langsamen Peripheriegeräten, wie beispielsweise Diskettenlaufwerken und Druckern.

Das Compilieren erfolgt bei beiden Pascal-Versionen sehr schnell. Pascal 80 kann aber nicht direkt im Speicher übersetzen. Gegen die speicherresidente Compilierung von Turbo-Pascal geht das Schreiben auf Diskette natürlich sehr langsam. Da Turbo-Pascal (besonders bei den kleinen Schneider-Modellen) nur wenig Platz für das Programm zur Verfügung stellt, spielt diese Einschränkung aber nur eine kleine Rolle.

Pascal80 besitzt neben den Standardbefehlen einige zusätzliche vordefinierte Prozeduren und Funktionen. So kennt Pascal80 die Funktion ENTIER, die der Basic-Funktion INT entspricht.

Mit INLINE werden Maschinencode-Bytes in das Pascal-Programm eingefügt. Erfreulicherweise arbeitet Pascal 80 auch mit hexadezimalen Zahlen. Im Gegensatz zu Turbo-Pascal werden diese nicht mit dem Dollar-Zeichen (»$«), sondern mit dem Doppelkreuz gekennzeichnet (»#«). Zu den weiteren maschinennahen Erweiterungen zählen USER (Aufruf von Maschinencode-Routinen), PEEK, POKE, INP und OUT. HALT bricht ein laufendes Programm ab und gibt den Stand des Programmzählers auf dem Bildschirm aus.

## CP/M mit Dummy

PRON lenkt die Bildschirmausgabe auf den Drucker um, PROFF macht das rückgängig. Diese Technik hat sowohl Vor- als auch Nachteile im Vergleich zum Turbo-Pascal-Befehl WRITE(LST).

Turbo-Pascal kennt zum Aufruf des CP/M-Betriebssystems die Funktion BDOS, die unter gleichem Namen auch als Prozedur vorhanden ist:

```
Bdos(13);
A:=Bdos(32);
```

In Pascal80 heißt diese Anweisung CPM und ist nur als Funktion vorhanden. Gegebenfalls muß ein Dummy-Parameter eingefügt werden. Ebenso muß etwas umständlich stets der Wert des DE-Registers angegeben werden:

```
Dummy:=CPM(14,2);
Dummy:=CPM(13,Dummy);
```

Weitergehende Befehle wie GOTOXY, ERASE, RENAME, FILEPOS oder FILESIZE unterstützt Pascal80 nicht. Ebenso fehlt der Datentyp STRING mit allen dazugehörenden Prozeduren und Funktionen wie COPY, CONCAT, LENGTH und POS.

Quasi als Entschädigung dafür hält die Diskette von Hisoft verschiedene Programmbibliotheken bereit, um die Grafik der Schneider-Computer anzusprechen. Für die CPC-Geräte sind die Dateien TURTLE2.PAS und TURTLE3.PAS gedacht, die die auf dem CPC vorhandenen ROM-Routinen ausnutzen.

Ein besonderes Bonbon ist aber die Darstellung von GSX-Grafiken. GSX heißt »Graphics System Extension« und bereichert den BDOS-Befehlsumfang um einen Aufruf zur geräteunabhängigen Grafikausgabe. Damit erscheint eine Grafik auf dem Bildschirm genauso wie auf dem Drucker oder Plotter. Auch der Austausch von Grafiken zwischen verschiedenen Computern ist damit möglich. GSX arbeitet nur unter CP/M Plus und ist deshalb ausschließlich dem CPC 6128 und dem Joyce vorbehalten.

GSX ist zwar langsamer als die Grafik über Routinen des Betriebssystems, dafür aber ungleich leistungsfähiger. So können mit jeweils einem einzigen Funktionsaufruf Flächen mit verschiedenen Mustern gefüllt oder Vielecke gezeichnet werden.

Am Anfang dieses Artikels wurde erwähnt, daß ED80 und HP beziehungsweise HP80 getrennte Programme sind, und deswegen Hisoft-Pascal80 nach dem altmodischen Prinzip getrennter Editierung und Compilierung arbeitet. Das ist auch richtig. Allerdings besitzt die Version Pascal 80V2 zusätzlich eine verbesserte Benutzeroberfläche. Sie trägt den Namen HPE.COM und macht Pascal80 Turbo-Pascal ähnlich. HPE vereint den Editor und den Compiler unter einem »gemeinsamen Dach«.

Ähnlich wie unter Turbo-Pascal ruft man durch Drücken einzelner Tasten verschiedene Programmteile auf. Das Hauptmenü sieht folgendermaßen aus:

```
HiSoft Pascal80 Menu Selection

(S)tart Editing
(C)ompile
(R)un
e(X)ecute
(Q)uit

(E)dit File
(M)ain File
```

Den unter Turbo-Pascal im Menü stehenden Punkt »Find Runtime-Error« hat Hisoft allerdings unter den Compiler-Optionen versteckt – auf daß niemand ihn finde.

So schön der Betriebssystem-Aufsatz HPE auch ist, er bleibt als nachträglich »übergestülpt« erkennbar. Denn HPE muß den Editor und den Compiler immer von der Diskette nachladen.

Der Vorteil dieser Methode ist, daß eine Menge Speicherplatz für das Quellprogramm freibleibt. So läßt sich auch in der Betriebssystem-Version CP/M 2.2 vernünftig mit Pascal80 arbeiten – was sich wegen Speicherplatzmangel von Turbo-Pascal nicht behaupten läßt. Der freie RAM-Bereich, der beim ED80 rund 30 KByte beträgt, macht bei Turbo-Pascal unter CP/M 2.2 ohne Speichererweiterung nur noch 6 bis 8 KByte aus – nicht gerade üppig.

## Handbuch auf Englisch

Bevor man anfängt, Programme für Turbo-Pascal in Include-Dateien zu zerstückeln, ist die Implementation von Pascal80 sinnvoller.

Das Handbuch von Hisoft mindert den alles in allem recht positiven Eindruck von Pascal80 ganz erheblich.

Es ist (für ein englisches Programm natürlich) in englischer Sprache gehalten. Im Gegensatz zum Turbo-Pascal-Handbuch, mit dem man auch Pascal

selbst lernen kann, listet es die Sprachelemente nur kurz auf und begnügt sich oft sogar mit Syntaxdiagrammen.

Besonders ärgert das Loseblatt-System des Handbuchs, das Hisoft dazu verleitet hat, bei jeder neuen Compiler-Version lediglich einige Ergänzungsseiten beizuheften. So erfährt der Leser auf der ersten Seite, wie er auf dem Joyce eine Backup-Kopie der neuesten Pascal80-Version erstellen kann. Auf der Rückseite dieses Blattes findet sich der Hinweis, daß in der Version vom 30. Mai 1986 die Prozedur CHAIN implementiert wurde – dabei vertreibt Hisoft längst das Nachfolgeprogramm.

Nachträglich eingefügt wurden auch das ausführliche Kapitel über die GSX-Grafik und weitere Informationen über die – jetzt wirklich neueste – Version des Compilers. Ganz hinten im Ordner entdeckt man noch einige Einlegeblätter, die die eigentlich sehr wichtige Benutzeroberfläche HPE beschreiben.

Alles in allem ist die Dokumentation kein Meisterwerk. Überarbeitung des wirren Konzepts ist dem Benutzer zuliebe dringend angeraten.

### Turbo oder Pascal80?

Falls Sie sich trotz der genannten Fakten und Meinungen noch nicht für einen der beiden Compiler entscheiden konnten, lassen Sie uns das Testergebnis auf einen Nenner bringen.

Turbo-Pascal ist allen zu empfehlen, die genügend freien Speicherplatz haben. Das kann entweder eine Speichererweiterung für CP/M 2.2 sein oder das »große« Betriebssystem CP/M Plus. Hier kann man es sich durchaus leisten, gleichzeitig den Compiler und den Editor im Speicher zu halten. Für Turbo-Pascal spricht weiter die vielfältige Literatur, die von den verschiedensten Verlagen angeboten wird. Hier sind andere Pascal-Versionen ganz klar im Nachteil.

Wer hingegen nur über den mageren Speicher von CP/M 2.2 auf dem CPC 464 oder CPC 664 herrschen kann und dennoch in Pascal programmieren will, für den steht Pascal80V2 ganz vorne. Daneben mag noch der Preisvorteil (39,95 Pfund, das sind zirka 120 Mark für Hisoft-Pascal80V2 gegenüber 225 Mark für die billigste Version von Turbo-Pascal) den Ausschlag geben. (Martin Kotulla/hg)

# Basic-Alternativen

**Auch der Urahn aller Basic-Interpreter, Microsoft-Basic, wird seit einiger Zeit in einem Paket mit passendem Compiler und Assembler für weniger als 200 Mark verkauft. Was bietet dieser bewährte Interpreter im Vergleich mit dem Locomotive-Basic der Schneider-Computer?**

MBasic – selten hat ein Softwareprodukt den Computermarkt so stark beeinflußt und unübersehbare Spuren hinterlassen wie dieser Basic-Interpreter von Microsoft. Nicht einmal Turbo-Pascal, das den Durchbruch für die Programmiersprache Pascal brachte, zeigte eine solch nachhaltige Wirkung. Eine ganze Generation von Programmierern wuchs mit MBasic oder einer seiner zahlreichen Ableger auf.

Mitte der siebziger Jahre waren Computer für viele Menschen einfach nur große undurchschaubare hochtechnische Wunderkisten, vor denen man entweder in Ehrfurcht erstarrte oder die man ob ihres unheilvollen Wirkens bekämpfte. Computer waren in Rechenzentren »eingesperrt« und für Normalsterbliche nur selten zugänglich.

Doch langsam aber stetig wuchs die Schar der Computerbegeisterten, die die »Bastelei« an diesen Maschinen zu ihrem Hobby machten. Aufgrund der mageren Speicherkapazitäten der kleinen Brüder der Großrechenanlagen – vier KByte waren unvorstellbar viel – programmierte man die »Gerätchen« meist in Maschinensprache.

Assemblersprachen setzen aber enorme Kenntnisse voraus. Deshalb waren bald Interpreter für die Programmiersprache Basic, die sich als besonders leicht erlernbar herausgestellt hatte, der »letzte Schrei«. Eine Version nannte sich noch verschämt »Tiny-Basic«, da ihr Sprachumfang recht mager war.

### Die Superidee des Bill Gates

Auch ein damals noch Jugendlicher namens Bill Gates versuchte sich an einem Basic-Interpreter. Und dieser war in seiner Zeit so gut, daß er der Vorläufer des heutigen Microsoft-Basic wurde. Schon die ersten Versionen von MBasic fanden gewaltigen Anklang bei Hobbyfreunden und auch professionellen Programmierern. Als logische Konsequenz seines Erfolges verlegte sich Bill Gates ganz auf das Programmieren und gründete die Firma Microsoft. Der Rest der Geschichte ist wohlbekannt – Microsoft wuchs und wuchs. Als sich der Computerriese IBM Anfang der achtziger Jahre entschied, von Microsoft ein Betriebssystem unter dem Namen MS-DOS einzukaufen und auf PC-DOS umzutaufen, war der Weg an die Spitze nicht mehr aufzuhalten.

Sieht man sich Programme wie MS-Word, MS-Pascal und QuickBasic oder das regelmäßig in neuen Versionen erscheinende Betriebssystem MS-DOS an, versteht man gut, daß für Microsoft der Markt der 8-Bit-Computer mit dem Betriebssystem CP/M keine Bedeutung mehr hat. Einfach »begraben« wollte man aber den Basic-Veteranen MBasic in der CP/M-Version wiederum auch nicht.

Die Konsequenz: MBasic, der Assembler Macro-80 und der Basic-Compiler Bascom sind als Paket jetzt zu einem auch für Hobbyanwender tragbaren Preis erschwinglich.

Microsoft-Basic ist sicherlich nicht das Nonplusultra heutiger Interpreter-Technologie. Als typisches CP/M-Basic kennt es beispielsweise keine Befehle zur direkten Ansteuerung des Bildschirms wie CLS, LOCATE oder WINDOW – von Grafikbefehlen ganz zu schweigen. Allenfalls unterstützt es TAB und POS. Man kann aber sowohl unter CP/M 2.2 als auch unter CP/M Plus über Escape-Sequenzen, die mit PRINT an den Bildschirm geschickt werden, fast alle bildschirmorientierten Befehle und Funktionen nachbilden. So ersetzt bei dem Betriebssystem CP/M 2.2 PRINT CHR$(12) den CLS-Befehl völlig. Aber auch auf Feinheiten wie einen bildschirmorientierten Editor oder zumindest den Copy-Cursor, wie beim Schneider-Basic, muß man verzichten. Der Programmierer muß mit EDIT eine Programmzeile in den Editier-

puffer holen und dann mit Hilfe verschiedener Tastenbefehle ändern. Dagegen ist es aber völlig problemlos, die Programme mit einem Textverarbeitungsprogramm zu bearbeiten und dann unter Basic mit LOAD zu laden.

## Großer Sprachumfang

Microsoft-Basic, manchmal auch Basic-80 genannt, ist ganz auf kommerzielle und wissenschaftliche Anwendung ausgerichtet. Das zeigt sich insbesondere daran, daß Berechnungen sowohl mit einfacher Rechengenauigkeit (fünf Nachkommastellen) als auch mit doppelter durchgeführt werden können. Dann besitzen Zahlen eine Genauigkeit von bis zu 15 Stellen nach dem Dezimalpunkt. Ein Komfort, vor dem viele neuere Basic-Dialekte passen müssen.

Die MBasic-Funktionen wie SIN, COS, LOG und SQR erkennen schon am Variablentyp, welche Genauigkeit gefordert ist.

```
A=SQR(2)     1.41421
A#=SQR(2)  1.414213538169861
```

Zur Umwandlung von Zahlen zwischen beiden Typen dienen die Funktionen CDBL und CSNG. Der Datentyp Integer ist ebenfalls vorgesehen. Er hat immer eine Breite von 16 Bit.

Sequentielle Dateien werden mit OPEN, CLOSE, INPUT#, LINE INPUT#, PRINT# und WRITE# bearbeitet. Als Besonderheit gegenüber dem Schneider-Basic kennt Basic-80 die Verwaltung von Dateien im Direktzugriffsverfahren. Die dazu notwendigen Prozeduren sind aber recht umständlich zu handhaben. So muß man den Aufbau jedes Datensatzes mit FIELD definieren. Danach werden Daten mit GET gelesen und mit PUT geschrieben. Zahlenwerte liest MBasic als Strings ein, sie müssen nachträglich explizit mit CVI, CVS oder CVD - je nach Zahlentyp - in numerische Werte umgesetzt werden. Beim Schreiben numerischer Daten muß dieser Vorgang mit MKI$, MKS$ und MKD$ umgekehrt durchgeführt werden. Ansonsten entspricht der Sprachumfang von MBasic weitgehend dem heutigen Basic-Interpreter im Heim- und PC-Bereich.

AUTO, RENUM und EDIT vereinfachen die Programmeingabe; TRON und TROFF helfen bei der Fehlersuche; COMMON und CHAIN erlauben die Verkettung von Basic-Programmen einschließlich Übergabe von Variablenwerten. Damit dürfen Basic-Programme unabhängig vom Speicherbereich soweit anwachsen, wie es die Diskettenkapazität zuläßt.

Auch CP/M-Standardbefehle wie ERA, DIR und REN sind in den Interpreter eingebaut - allerdings unter anderem Namen. So gibt FILES das Inhaltsverzeichnis der Diskette aus, KILL löscht Dateien und NAME-AS benennt sie um:

```
NAME "DATEI" AS "OLDFILE"
```

Auf der maschinennahen Ebene glänzt Microsoft-Basic ebenfalls mit einer sehr guten Ausstattung. Maschinencode-Programme werden entweder mit CALL oder mit USR aufgerufen. Dabei ist die Adresse von USR allerdings vorher mit DEF USR festzulegen. PEEK und POKE gehören zum Standard eines Basic-Interpreters WAIT, INP und OUT zur Behandlung der Prozessorports hingegen sind außergewöhnlich.

Auf der Diskette von MBasic finden Sie noch den Vorgänger mit dem Namen »OBasic«. Warum denn das, fragen Sie jetzt sicher. OBasic (Basic-80, Version 4.51) kennt nur Variablennamen mit maximal zwei Buchstaben Länge. Zum Trennen der Befehle muß kein Leerzeichen vorhanden sein.

```
IFA=3THENPRINT"ENDE":END
```

ist für OBasic damit eine syntaktisch korrekte Befehlszeile. MBasic (Version 5.21 vom 28. Juli 1981) meldet hingegen einen Fehler.

Da aber eine Reihe von alten Basic-Programmen, zum Beispiel unter Public-Domain, existieren, die keine Leerzeichen zwischen den Befehlen enthalten, laufen diese unter MBasic nicht. Hier kommt OBasic zum Einsatz. Ein netter Nebeneffekt: Unter OBasic stehen fast sieben KByte mehr Speicherplatz zur Verfügung als unter MBasic.

## Ein schneller Compiler...

Das interessanteste Programm auf der Microsoft-Basic-Diskette ist wohl der Compiler. Durch die Kombination von Interpreter und Compiler in einem Paket ist es möglich, mit dem Interpreter Programme interaktiv zu entwickeln und bis zur völligen Fehlerfreiheit auszutesten. Später setzt man sie dem Compiler BASCOM vor, der daraus lauffähige CP/M-Dateien erzeugt. Solche Programme kann man dann auch an andere Computerbesitzer weitergeben, die kein MBasic besitzen. Die Abarbeitungsgeschwindigkeit des compilierten Programms ist zudem zirka drei- bis zehnmal höher.

Was ist das Schöne an einem Basic-Compiler? Er überträgt ohne große Schwierigkeiten ein im Interpretermodus geschriebenes Programm in puren Maschinencode. Und daran kranken ja bekanntlich die meisten Basic-Compiler unter Amsdos auf dem Schneider. Abgesehen von wirklich feinen Unterschieden, etwa bei dem internen Speicherformat von Variablen oder bei den Befehlen EDIT und RENUM in compilierten und interpretierten Programmen, arbeitet ein MBasic-Programm auch in compilierter Form problemlos. Sogar Fließkommazahlen, bei denen viele Compiler das Handtuch werfen, verarbeitet BASCOM.

## ...und ein toller Assembler

Auf der CP/M-Systemdiskette von Schneider finden Sie bereits einen kompletten Assembler mit dem Namen ASM. CP/M Plus besitzt sogar zwei Assembler: MAC und RMAC. Doch die Programme von Digital Research verarbeiten ausschließlich Mnemonics des Prozessors Intel-8080. Der Microsoft-Assembler Macro-80, kurz M80 genannt, läßt sich im Gegensatz dazu mit den Assembler-Direktiven ».8080« und ».Z80« zwischen beiden Maschinensprachen umschalten. So kann jeder nach seinen Präferenzen programmieren.

Ferner ist M80 vollständig makrofähig und erzeugt linkfähigen Code. Dabei werden Labels über die Direktiven EXTERNAL und PUBLIC auch der Außenwelt - sprich anderen Programmteilen - bekanntgemacht. Der Objektcode liegt im Microsoft-.REL-Format vor und ist damit voll relokatibel. Der Linker L80 entscheidet erst über die benutzten Speicheradressen.

Zu Macro-80 gehören die beiden Utilities CREF80.COM und LIB. CREF80 erzeugt aus dem Assemblercode ein Querverweis-Listing. LIB steuert den Aufbau und die Verwaltung von Programmbibliotheken. MBasic mit Macro-80 und Bascom gegen Locomotive-Basic. Welche Kombination ist besser: Der moderne Interpreter von Locomotive oder das »Relikt« aus den siebziger Jahren«?

Zugegeben, etwas ungewöhnlich ist die Kombination von Basic-Interpreter, Compiler und Assembler samt Utilities schon - aber unheimlich praktisch. Wer nur in Basic programmieren will, benutzt nur MBasic mit Compiler, der Maschinensprache-Fan dagegen gibt natürlich M80 und L80 den Vorzug. Wer Maschinencode-Routinen und Basic-Programme kombinieren will, schöpft dann alle Programme auf der Diskette voll aus. Wer das Mallard-Basic auf dem Joyce besitzt, für den ist der Compiler besonders interessant. MBasic ist nämlich eine Untermenge des Mallard-Basic.

(Martin Kotulla/hg)

# Fixe Daten

**Leistungsfähige Datenverwaltung beginnt erst mit dem relativen Zugriff. Doch von Haus aus bieten die CPCs solche Fähigkeiten nicht.**

Datenverwaltung gehört zu den Dingen, für deren Erledigung der Computer wie geschaffen ist. Er hilft Ihnen schnell und präzise, sich dieser Fleißarbeit zu entledigen. Ob Sie personenbezogene Daten wie beispielsweise Adressen, Umsätze und Geburtstage speichern wollen oder andere Informationen wie die Musiktitel Ihrer Schallplattensammlung – der Nutzen ist groß. Bei der programmtechnischen Umsetzung eines solchen Vorhabens stehen zwei Verfahren zur Wahl. Die sogenannte sequentielle Verarbeitung bringt den Vorteil einer hohen Geschwindigkeit. Ihn erkauft man sich jedoch mit kleiner Kapazität. Daher eignet sich diese Art der Verarbeitung vor allem für kleinere Datenbestände. Wie der Name schon sagt, erfolgt die Speicherung »aufeinanderfolgend«. In der Praxis wirkt sich das so aus, daß der gesamte Datenbestand einer Datei einmalig in den Arbeitsspeicher des Computers geladen wird. Dort lassen sich die Daten durch den Benutzer bearbeiten und manipulieren, bis der Computer auf Kommando sämtliche Daten wieder en bloc auf dem Massenspeichermedium sichert.

Ganz anders der »relative« Zugriff. Hier befinden sich die Daten einer Datei zu keinem Zeitpunkt gemeinsam im RAM. Vielmehr steht zur augenblicklichen Bearbeitung stets nur ein einziger Datensatz bereit. Der Computer sucht sich also innerhalb der Diskettendatei (aufgrund des nötigen direkten Zugriffs ist der Kassettenrecorder als Medium ungeeignet) den gewünschten Datensatz und lädt ihn. Nach Bearbeitung speichert er ihn sofort wieder, um sich dem nächsten zuzuwenden. Die Vorteile liegen auf der Hand:

– Die Größe der Datei begrenzt nun nicht mehr der Arbeitsspeicher des Computers, sondern die viel umfangreichere Kapazität der Diskette.

– Bei Stromausfall, »Systemabsturz« oder Bedienungsfehlern ist – wenn überhaupt – mit dem zuletzt bearbeiteten nur ein einziger Datensatz verloren.

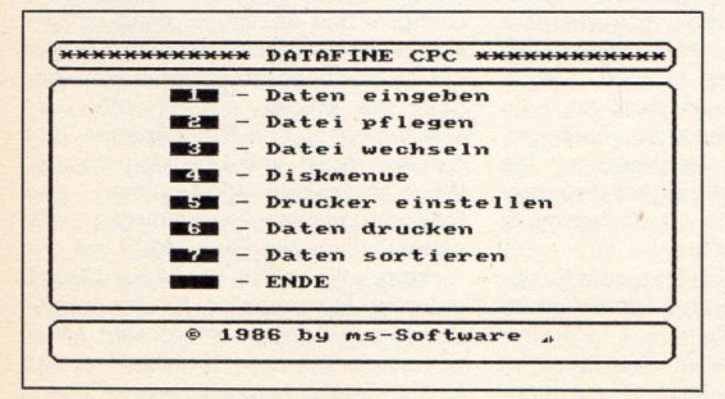
– Zum Sortieren braucht man nicht den gesamten Speicherinhalt umzuschichten, sondern benutzt die vorhandene Numerierung zur Bildung eines Index.

Wer mit seinen Überlegungen zu genau diesen Schlußfolgerungen kommt, erlebt eine herbe Enttäuschung, wenn er erfährt, daß sein CPC ihn in seinem Vorhaben, diesen Wunsch in die Realität umzusetzen, nicht unterstützt. Im Gegenteil, das DOS der ansonsten doch eher leistungsfähigen 3-Zoll-Diskettenlaufwerke bietet dem Basic-Programmierer keine Möglichkeit zum direkten Datenzugriff (für Besitzer eines Vortex-Laufwerks gilt die Aussage freilich nicht). Und so hilft nur, dem Basic-Programm mit einer eigenen kleinen Maschinencode-Routine hilfreich unter die Arme zu greifen. Diese Routine erlaubt mit Hilfe zweier RSX-Befehle eben das, wozu unser DOS nicht in der Lage ist.

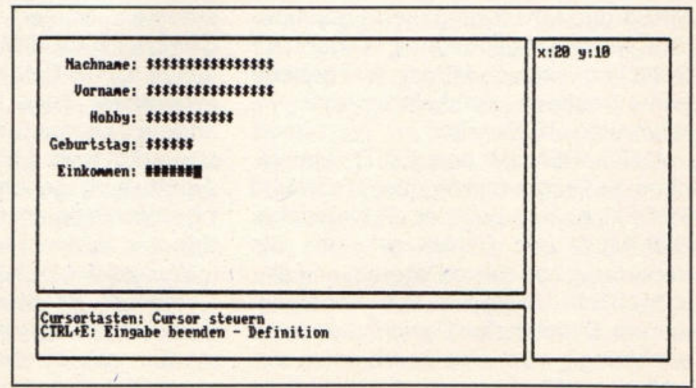
Für »Datafine« erfüllt Listing 1 diesen Zweck. Wenn Sie es eingegeben und gestartet haben, erzeugt es automatisch auf Diskette die Binärdatei »DATAFINE.BIN«. Das Hauptprogramm (Listing 2) lädt den Maschinencode und bindet die RSX-Befehle ein. Da Datafine voll menügesteuert ist, sind hier nur ein paar wichtige Ergänzungen zu einzelnen Punkten genannt: Bei der ersten Inbetriebnahme wählen Sie bitte zunächst einmal den Punkt 1 (Daten eingeben). Dort veranlassen Sie Datafine zur Neuanlage Ihrer Datei. Alle anderen Funktionen arbeiten nämlich nur mit bereits bestehenden Dateien. Auf die Frage nach der Zahl der Datensätze antworten Sie mit Eingabe einer Zahl zwischen 1 und 999. Diese Beschränkung auf ein Maximum von 1000 Sätzen erfolgt in Zeile 2630. Abhängig von der Diskettenkapazität auf der einen Seite und dem Umfang der Datensätze auf der anderen, lassen sich hier auch größere Werte eintragen. Sie müssen dann nur ausprobieren, bei welcher Dimension Datafine mit einer Fehlermeldung wegen mangelnder Kapazität abbricht. Natürlich existieren bei diesem ersten Lauf auch noch keine vordefinierten Eingabemasken. Also legen Sie das Bildschirmformat für die spätere Bearbeitung fest. Dazu bewegen Sie den Cursor mit den dazu vorgesehenen Tasten. An der gewünschten Position geben Sie über die Tastatur zuerst die Feldnamen ein. Sie dürfen keinesfalls Leerzeichen enthalten und müssen mit dem Doppelpunkt enden. Hinter dem Doppelpunkt folgt nach einem Zwischenraum (Leertaste) das Datenfeld. Hier legen Sie mit den beiden Zeichen <#> und <$> die Eingabelänge und Art der Daten fest. Für alphanumerische (gemischte) Texte wählen Sie das <$>, während <#> für rein numerische (Zahlen-)Felder steht. Die jeweilige Anzahl der Zeichen bestimmt die Feldlänge.

```
Name: $$$$$$$$$$$$$$$$$$$$
PLZ:  ####
```

In unserem Beispiel erlauben wir im Feld »Name« die Eingabe von 20 Buchstaben oder beliebigen anderen ASCII-Zeichen. Für die Postleitzahl (»PLZ«) akzeptiert der Datafine jedoch nur einen Zahlenwert aus maximal vier Ziffern. Beim Aufbau Ihrer Maske sind Sie in der Wahl des Formats kaum eingeschränkt: Die einzugebende Zahl der Zeichen darf pro Datensatz 255 nicht überschreiten, mehr als 20 verschiedene Felder verwaltet die Maske nicht, und Felder dürfen sich immer nur über eine Bildschirmzeile erstrecken. Wenn Sie mit Ihrem Werk zufrieden sind, leiten Sie mit der Tastenkombination <CTRL-E> das Speichern der Maske ein. Da sie in einer eigenen Datei gespeichert ist, läßt sich die Maske jederzeit verändern oder für andere Dateien verwenden. Nach der Maskengestaltung kontrolliert sie der Computer auf Fehler, um dann gegebenenfalls in den Editiermodus zurückzukeh-




Das übersichtliche Hauptmenü




Eingabemasken nach Ihren Wünschen

ren. Ursachen für solche Fehler können sein:

- Fehlerhafte Syntax (beispielsweise die Indikatoren »$« und »#« innerhalb eines Feldes gemischt oder das Leerzeichen hinter dem Doppelpunkt vergessen).
- Felder zu groß.
- Zu viele Felder innerhalb eines Datensatzes.

Zur Eröffnung neuer Dateien legt Datafine die komplette (leere) Datei bereits auf Diskette an, was einige Zeit in Anspruch nimmt. Erst danach dürfen Sie mit Eingabe Ihrer ersten Daten beginnen. Diesen Modus brechen Sie mit der COPY-Taste ab. Der Menüpunkt »Datei pflegen« wird der meistgebrauchte, wenn Sie erst einmal alle Daten erfaßt haben. Zur Funktion benötigt er auf der Datendiskette mindestens drei Dateien. »NAME.« enthält die gespeicherten Daten, die Indexdatei »NAME.IND« dient der Verwaltung der Daten, und »NAME.MSK« definiert die Eingabemaske.

Innerhalb der Dateipflege lassen sich die Daten korrigieren, löschen, suchen, anzeigen und drucken. Im Blättermodus gelangen Sie durch die Cursor-Steuertasten <Cursor-auf> und <Cursor-ab> in den nächsten beziehungsweise vorhergehenden Datensatz. Wollen Sie Korrekturen vornehmen, bringen Sie das invers dargestellte Korrekturfeld mit den Cursortasten in Position und drücken dort <ENTER> (<RETURN>). Erst dann überschreiben Sie das gewünschte Feld mit neuen Daten. Die Suche erfolgt mit einem höchstens 35 Zeichen langen Suchbegriff und erstreckt sich auf sämtliche Datenfelder. Die zu durchsuchenden Datensätze lassen sich beschränken.

Wichtig vor der ersten Druckausgabe ist der Aufruf der Druckerinitialisierung, um das genaue Format dafür festzulegen. Das Diskettenmenü umfaßt nützliche Routinen wie den Wechsel des Bezugslaufwerks (A oder B) oder das Löschen und Umbenennen von Dateien. Aber hier läßt sich auch in andere User-Bereiche umschalten, um beispielsweise mehrere kleine Dateien auf einer Diskette deutlich voneinander zu trennen.

Die Druckerinstallation ist ein wenig aufwendiger, erlaubt aber auch eine sehr flexible Anpassung der Druckroutinen an die jeweiligen Anforderungen – ob Sie nun Adreßaufkleber, Kundenlisten oder nur die Werte Ihrer erfaßten Sammlerstücke brauchen. Zuerst wählen Sie aus, zu welchen Feldern die Feldnamen mitzudrucken sind. Danach legen Sie für die einzelnen Feldinhalte fest, ob, wo und in welcher Reihenfolge sie Datafine auf Papier ausgeben soll. Drücken Sie einfach <ENTER>, überspringt später die Druckroutine dieses Feld. Zur Formatierung des Druckbilds benutzen Sie die Cursor-Steuertasten. Ein nach rechts gerichteter Pfeil hängt das Feld direkt an das vorhergehende an. Ein abwärts zeigender Pfeil bewirkt ein Linefeed und damit den Druck in der nächsten Zeile. Benutzen Sie die Pfeile mehrfach, bedeutet das die entsprechende Zahl von Zeilenvorschüben.

Für die Sortierung wählen Sie ein beliebiges Feld. Auf gar keinen Fall dürfen Sie jemals während eines Sortiervorgangs die Diskette aus dem Laufwerk nehmen, denn das zerstört Ihre Datei.

Mit der COPY-Taste beenden Sie die Vorgänge »Drucken«, »Sortieren« und »Eingabe« und kehren ins Hauptmenü zurück. (Michael Straßer/ja)

| Steckbrief | |
|---|---|
| Programm: | Datafine |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora/CPC |
| Datenträger: | Diskette |
| Besonderes: | arbeitet nur mit Amsdos |

```
100 '*************************************  [31D4]
101 '*DATAFINE.DAT - DATA-Lader von 'CPC'*  [A986]
102 '*************************************  [A3D8]
103 '                                       [DEB6]
104 DATA A000,01,0A,A0,21,22,A0,CD,D1,16CB  [C75E]
105 DATA A008,BC,C9,12,A0,C3,CF,A0,C3,60A7  [09DC]
106 DATA A010,0D,A1,D2,D7,00,00,00,00,39F0  [2310]
107 DATA A018,00,00,00,00,00,00,00,00,0000  [9630]
108 DATA A020,00,00,FC,A6,0A,A0,00,00,1730  [D5FE]
109 DATA A028,00,00,00,00,00,00,00,00,0000  [FC36]
110 DATA A030,00,C3,CF,A0,C3,0D,A1,FE,24B0  [FBB6]
111 DATA A038,02,C2,B4,A0,DD,6E,02,DD,2A89  [E0C8]
112 DATA A040,66,03,2B,7E,FE,04,C2,B4,3790  [3776]
113 DATA A048,A0,23,11,26,A0,01,05,00,5D8E  [AB0A]
114 DATA A050,ED,B0,DD,6E,00,DD,66,01,4479  [3ABA]
115 DATA A058,2B,7E,FE,02,C2,B4,A0,23,1043  [EC82]
116 DATA A060,11,2E,A0,01,03,00,ED,B0,1662  [1E1A]
117 DATA A068,3A,2A,A0,FE,81,30,0D,21,0893  [5770]
118 DATA A070,00,00,22,2B,A0,3E,00,32,033A  [E2EA]
119 DATA A078,2D,A0,37,C9,FE,98,D2,B4,3070  [CFC0]
120 DATA A080,A0,21,29,A0,CB,FE,3A,2A,529E  [F0BC]
121 DATA A088,A0,FE,97,28,0F,21,2A,A0,7FE8  [6FD0]
122 DATA A090,34,2B,CB,3E,2B,CB,1E,2B,0823  [26BA]
123 DATA A098,CB,1E,18,EA,2A,28,A0,22,6F32  [BFB6]
124 DATA A0A0,2B,A0,3A,27,A0,CB,3F,32,3ED0  [AFC2]
125 DATA A0A8,2D,A0,3A,2E,A0,FE,00,CA,3D12  [06E8]
126 DATA A0B0,B4,A0,37,C9,37,3F,C9,2A,788C  [04D6]
127 DATA A0B8,2B,A0,23,22,2B,A0,11,29,3813  [1334]
128 DATA A0C0,00,2A,7D,BE,19,3A,2B,A0,0E16  [38B2]
129 DATA A0C8,77,23,3A,2C,A0,77,C9,CD,3343  [B7B2]
130 DATA A0D0,37,A0,D0,CD,DC,A0,CD,F8,2052  [99F6]
131 DATA A0D8,A0,ED,B0,C9,CD,BE,A0,21,7421  [2F00]
132 DATA A0E0,80,A1,CD,5A,A1,CD,B7,A0,73B2  [21F4]
133 DATA A0E8,21,00,A2,CD,5A,A1,CD,B7,0969  [C4D0]
134 DATA A0F0,A0,21,80,A2,CD,5A,A1,C9,44EB  [07DC]
135 DATA A0F8,3A,2D,A0,5F,16,00,21,80,07C2  [C964]
136 DATA A100,A1,19,ED,5B,2F,A0,3A,2E,4D72  [DBC6]
137 DATA A108,A0,4F,06,00,C9,CD,37,A0,46B2  [6D90]
138 DATA A110,D0,CD,BE,A0,21,80,A1,CD,4407  [6EB6]
139 DATA A118,5A,A1,CD,F8,A0,EB,ED,B0,14A6  [8C34]
140 DATA A120,CD,BE,A0,21,80,A1,CD,61,586F  [CFC4]
141 DATA A128,A1,CD,B7,A0,21,00,A2,CD,7FA1  [BCCE]
142 DATA A130,5A,A1,CD,F8,A0,EB,ED,B0,14A6  [011C]
143 DATA A138,CD,BE,A0,21,00,A2,CD,61,5C63  [F7BE]
144 DATA A140,A1,CD,B7,A0,21,80,A2,CD,7DA1  [D9D4]
145 DATA A148,5A,A1,CD,F8,A0,EB,ED,B0,14A6  [DA34]
146 DATA A150,CD,BE,A0,21,80,A2,CD,61,5863  [42B2]
147 DATA A158,A1,C9,DF,5E,A1,C9,92,D3,7B1B  [A41E]
148 DATA A160,07,DF,65,A1,C9,68,A1,07,345D  [8398]
149 DATA A168,E5,D5,C5,E5,11,08,00,CD,5155  [7F94]
150 DATA A170,98,CA,CD,10,D4,D2,A9,D3,6249  [86CC]
151 DATA A178,EB,E3,CD,F3,D9,C3,A6,D3,5F8B  [ED66]
152 DATA *ENDE*                             [88C4]
153 adr=&A000:zeile=104:MEMORY &9FFF        [FB5A]
154 READ d$:IF d$="*ENDE*"THEN 165          [F296]
155 pr=0                                    [1C14]
156 FOR i=1 TO 8                            [016A]
157 READ a$:a=VAL("&"+a$)                   [A448]
158 POKE adr,a:adr=adr+1                    [5124]
159 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [B4A4]
160 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                       [EDAA]
161 NEXT i                                  [3B00]
162 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                          [CC8C]
163 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                   [1F14]
164 zeile=zeile+1:GOTO 154                  [C966]
165 SAVE"DATAFINE.BIN",B,&A000,&180:END     [32FE]
```

**Listing 1. Ein Basic-Lader erzeugt den Maschinencode der RSX-Befehlserweiterung**

```
10  ' ****************                      [447C]
20  ' * DATAFINE     *                      [5EF2]
30  ' * (c) 1986 by *                       [2E0A]
40  ' * ms-Software *                       [DD6E]
50  ' * M. Strasser *                       [21A4]
60  ' * Rottalstr.5 *                       [5926]
70  ' *8 Muenchen 80*                       [C46A]
80  ' ****************                      [6D8A]
90  ' V1.4 vom 6.9.1986                     [8CDC]
100 ' ************************************
    ******
                * Programm DATAFINE.BIN sc
    hon geladen? *
                        ******************
    ***********************             [777C]
110 IF PEEK(&A000)=&1 AND PEEK(&A001)=&A
     THEN 190                              [7120]
120 ' ** Nein, dann bitte laden **

                ** Ja, dann ab zum Menue
    **                                     [319C]
130 MEMORY 39999:LOAD "DATAFINE.BIN",&A0
    00:CALL &A000:CLOSEIN                  [B938]
```

**Listing 2. Mit »Datafine« haben Sie Ihre Daten fest im Griff**

```
140 OPENOUT "d":MEMORY HIMEM-1:CLOSEOUT  [89A0]
150 DATA CD,60,BB,32,4A,9C,C9            [C576]
160 RESTORE 150                          [AB22]
170 FOR n=40000 TO 40006:READ a$:POKE n,
    VAL("&"+a$):NEXT                     [1240]
180 SYMBOL 255,136,216,175,152,152,142,1
    ,30:dinit=0 ' Druckerflag            [93CC]
190 DIM feld$(5),i$(5),a$(5),dr$(3):|A:|
    USER,0:user=0:drive$="A":INK 0,0:BOR
    DER 0:INK 1,24:INK 2,15:INK 3,6      [89C2]
200 GOTO 3760                            [05B6]
210 ' ******************

              * Daten eingeben *

                   ******************
                                         [4F7E]
220 CLS:korr=1                           [05F8]
230 PRINT#1,"*********** DATEN EINGEBEN
    ***********";                        [FBCC]
240 PEN 1:PRINT:PRINT" X 1 X in bestehen
    de Datei schreiben":PRINT"J X 9 X Ne
    ue Datei eroeffnen"                  [CCDE]
250 jn$="":WHILE jn$<>"1" AND jn$<>"9":j
    n$=INKEY$:WEND:IF jn$="9" THEN GOSUB
     2450:daz=0:GOTO 280                 [E606]
260 GOSUB 3290:IF dat$="" OR er=1 THEN 3
    760                                  [65A4]
270 IF daz<>0 AND daz=anz THEN daz=daz-1
    :GOTO 370                            [4ABC]
280 MODE 2:GOSUB 3500:OPENIN ""+dat$     [0584]
290 CLS:PRINT#2,"LDatei: ";dat$:PRINT#2,
    "JGroesse: "anz:PRINT#2,"JDaten: ";d
    az                                   [3FEA]
300 FOR n=1 TO feld:l=VAL("&"+LEFT$(feld
    $(n),2)):t=VAL("&"+MID$(feld$(n),3,2
    )):x=VAL("&"+MID$(feld$(n),5,2)):y=V
    AL("&"+MID$(feld$(n),7,2))           [970A]
310 feld$=RIGHT$(feld$(n),LEN(feld$(n))-
    8):LOCATE x,y:PRINT feld$;": ";:GOSU
    B 3400:IF ret THEN 390               [3A52]
320 i$(n)=in$:IF LEN(i$(n))<l THEN i$(n)
    =LEFT$(i$(n)+SPACE$(l),l)            [D274]
330 NEXT                                 [57EA]
340 PRINT#1,"LGJAlles richtig? (J/N)":jn
    $="":WHILE jn$<>"J" AND jn$<>"N":jn$
    =UPPER$(INKEY$):WEND:CLS #1:IF jn$="
    N" THEN num=daz:GOTO 630             [51D6]
350 a$="":FOR n=1 TO feld:a$=a$+i$(n):NE
    XT:po=daz*laenge                     [38EA]
360 |W,@po,@a$                           [AD90]
370 daz=daz+1:IF daz=anz THEN CLS:PRINT"
    JJJJJGDatei voll!!!":FOR n=1 TO 2000
    :NEXT:GOTO 390                       [9C4A]
380 GOTO 290                             [EF5E]
390 CLOSEIN:CLS:OPENOUT ""+dat$+".ind":P
    RINT#9,anz:PRINT#9,daz:CLOSEOUT:GOTO
     3760                                [587C]
400 ' ******************

              * Datei pflegen *


                    ****************** [8D96]
410 IF dat$="" OR er=1 THEN GOSUB 3290:I
    F dat$="" OR er=1 THEN 3760          [D2E2]
420 MODE 2:korr=2:GOSUB 3500             [AE6C]
430 PRINT#2,"******************Datei<2>pf
    legen******************"             [A14A]
440 PRINT#2,"JX "CHR$(240)" X - vor":PRI
    NT#2,"JX "CHR$(241)" X - zurueck"    [163A]
450 PRINT#2,"JX 1 X - Korrektur":PRINT#2
    ,"JX 2 X - Loeschen":PRINT#2,"JX 3 X
     - Nr.eingabe";                      [87B8]
460 PRINT#2,"JX 4 X - Suchen":PRINT#2,"J
    X 5 X - Drucken":PRINT#2,"JX 9 X - E
    NDE"                                 [4492]
470 fr=FRE(""):num=0:OPENIN ""+dat$      [E350]
480 PRINT#1,"LDateiname: ";UPPER$(dat$);
    TAB(35);USING "Groesse der Datei: ##
    ##";anz:PRINT#1,USING "Eingegebene D
    aten: ####";daz;:PRINT#1,TAB(35);USI
    NG "Freie Datensaetze: ####";anz-daz
                                         [A546]
490 PRINT#1,TAB(13);"DATAFINE CPC "CHR$(
    164)" 1986 by Happy-Computer"CHR$(25
    5)                                   [24AE]
500 CLS                                  [792E]
510 a$=SPACE$(laenge):po=num*laenge:|R,@
    po,@a$                               [4C8A]
520 GOSUB 3390:GOSUB 3250                [E994]
530 jn=ASC(INKEY$+CHR$(0))               [8C8C]
540 IF jn=240 OR jn=241 THEN 590 ELSE IF
     jn-48<1 OR jn-48>9 THEN 530         [A1E6]
550 jn=jn-48:IF jn>5 AND jn<9 THEN 530 E
    LSE IF jn=9 THEN GOTO 3760           [47B2]
560 ON jn GOTO 630,810,890,970,1130      [9062]
570 GOTO 570                             [C562]
580 ' *********************************
    *
              * Datensatz vor-/zurueckbl
    aettern *
                    ******************
    ******************                   [043E]
590 IF jn=240 THEN num=num+(1 AND num<da
    z-1)                                 [C49A]
600 IF jn=241 THEN num=num-(1 AND num>0)
                                         [2CBA]
610 GOTO 510                             [D24C]
620 ' *************************

              * Datensatz korrigieren *

                    ******************
    *******                              [5EEC]
630 PRINT#1,"LJ<2>Bewegen Sie mit "CHR$(
    240)" und "CHR$(241)" den X Pointer
    X, dann X>ENTER<X."                  [4EE6]
640 z=1                                  [3C44]
650 WHILE INKEY$<>"":WEND                [0E3C]
660 x=VAL("&"+MID$(feld$(z),5,2)):y=VAL(
    "&"+MID$(feld$(z),7,2))              [784A]
670 feld$=RIGHT$(feld$(z),LEN(feld$(z))-
    8)                                   [6F02]
680 LOCATE x,y:PRINT feld$;": X";i$(z);"
    X"                                   [71E8]
690 IF INKEY(0)>-1 THEN LOCATE x,y:PRINT
     feld$;": ";i$(z):z=z-(1 AND z>1):GO
    TO 660                               [708C]
700 IF INKEY(2)>-1 THEN LOCATE x,y:PRINT
     feld$;": ";i$(z):z=z+(1 AND z<feld)
    :GOTO 660                            [9C4C]
710 IF INKEY(18)>-1 OR INKEY(6)>-1 THEN
    CALL &BB03 ELSE GOTO 690             [D0B4]
720 l=VAL("&"+LEFT$(feld$(z),2)):t=VAL("
    &"+MID$(feld$(z),3,2))               [24FA]
730 LOCATE x,y:PRINT feld$;": ";         [EBD2]
740 GOSUB 3400:IF ret THEN CLS#1:ON korr
     GOTO 350,480                        [2512]
750 i$(z)=in$+SPACE$(l-LEN(in$))         [AB38]
760 PRINT#1,"LJ<2>Weiteres Feld korrigie
    ren? (J/N)J"                         [7C12]
770 jn$="":WHILE jn$<>"J" AND jn$<>"N":j
    n$=UPPER$(INKEY$):WEND               [A65A]
780 IF jn$="J" THEN 630                  [6A7A]
790 a$="":FOR n=1 TO feld:a$=a$+i$(n):NE
    XT:po=num*laenge:|W,@po,@a$:CLS #1:O
    N korr GOTO 370,480                  [D7EE]
800 ' ******************

              * Daten loeschen *

                    ******************
                                         [7DB0]
810 PRINT#1,"LJWollen Sie wirklich loesc
    hen? (J/N)G";:INPUT #1,jn$           [44BE]
820 IF UPPER$(jn$)="N" THEN 480          [5F80]
830 PRINT#1,"LX Datensatz"num"X wird gel
    oescht ";                            [4872]
840 IF num=daz-1 THEN a$=SPACE$(laenge):
    po=num*laenge:|W,@po,@a$:daz=daz-1:n
    um=num-1:PRINT:GOTO 870              [EFB2]
850 PRINT#1,"und mit X Datensatz"daz-1"X
    ":PRINT#1,"ueberschrieben."          [B37A]
860 a$=SPACE$(laenge):po=(daz-1)*laenge:
    |R,@po,@a$:po=num*laenge:|W,@po,@a$:
    daz=daz-1                            [5786]
870 OPENOUT ""+dat$+".ind":PRINT#9,anz:P
    RINT#9,daz:CLOSEOUT:GOTO 480         [0254]
880 ' ******************

              * Nummerneingabe *

                    ******************
                                         [B960]
890 CLS:PRINT"JJJJJ"CHR$(150)STRING$(23,
    154)CHR$(156)                        [955A]
900 PRINT CHR$(149)TAB(25)CHR$(149):PRIN
    T CHR$(147)STRING$(23,154)CHR$(153)  [F618]
910 PRINT"KKIIDatensatznummer: ";:l=4:t=
    2:GOSUB 3400                         [18BE]
920 IF ret THEN 500                      [6272]
930 in=VAL(in$):in=INT(in)               [6BC0]
940 IF in<0 THEN in=0 ELSE IF in>daz-1 T
    HEN in=daz-1                         [9B98]
950 num=in:PRINT"L":GOTO 510             [E2C4]
960 ' ****************

              * Datei suchen *

                    ****************  [BCCE]
970 CLS:PRINT"JJSuchbegriffG: ";:l=35:GO
    SUB 3400:IF ret THEN 480 ELSE su$=in
    $                                    [723A]
980 PRINT"JJSuchen von Datensatz ";:l=4:
    t=2:GOSUB 3400                       [01FA]
```

```
990 IF ret THEN 480 ELSE sua=VAL(in$):IF
    sua<0 THEN sua=0 ELSE IF sua>daz-1
   THEN sua=daz-1                         [29EC]
1000 PRINT"JJSuchen bis Datensatz ";:l=4
     :t=2:GOSUB 3400                      [5B10]
1010 IF ret THEN 480 ELSE sue=VAL(in$):I
     F sue<sua THEN sue=sua ELSE IF sue>
     daz-1 THEN sue=daz-1                 [FEB8]
1020 CLS #1                               [0572]
1030 FOR num=sua TO sue:a$=SPACE$(laenge
     ):po=num*laenge:|R,@po,@a$           [3574]
1040 LOCATE #1,1,2:PRINT#1,USING "Datens
     atz ####";num                        [CBC4]
1050 IF INSTR(a$,su$) THEN 1080           [FBD6]
1060 NEXT num:num=num-1                   [EB16]
1070 PRINT"JJGEnde der Suche":FOR n=1 TO
      2500:NEXT:GOTO 480                  [7D64]
1080 CLS:GOSUB 3390:GOSUB 3250            [CC30]
1090 PRINT#1,"LJWeitersuchen ?<2>(J/N)G"
                                          [820A]
1100 jn$="":WHILE jn$<>"J" AND jn$<>"N":
     jn$=UPPER$(INKEY$):WEND              [82A2]
1110 IF jn$="J" THEN CLS #1:GOTO 1060 EL
     SE 480                               [87FA]
1120 ' **********************************

          * Angezeigten Datensatz
      drucken *
                              *************
     ********************                 [3A3C]
1130 IF dinit=0 THEN CLS:PRINT"JJJJJJJGB
     itte erst X Drucker einstellen X !!
     ":FOR n=1 TO 2000:NEXT:GOTO 480      [A12E]
1140 GOSUB 3170                           [4BA2]
1150 num=num+(1 AND num<daz-1):GOTO 510   [88AC]
1160 ' ******************

          * Datei wechseln *
                              *************
     *****                                [1A16]
1170 CLS:PRINT"JJOBName der neuen Datei:
      OA";:l=8:GOSUB 3400:IF ret THEN 37
     60                                   [A6DA]
1180 er=0:dat$=in$:GOSUB 3660             [7A4C]
1190 IF er=1 THEN PEN 1:PRINT"JJGDatei n
     icht vorhanden":FOR n=1 TO 2000:NEX
     T:GOTO 1170 ELSE GOSUB 3330          [2826]
1200 dinit=0:GOTO 3760                    [5296]
1210 ' *************

          * Diskmenue *
                              *************
                                          [ABD0]
1220 MODE 2:GOSUB 3500:PRINT#2,"********
     *********<2>Disk-menue<2>**********
     *******"                             [B0E0]
1230 PRINT#2,"JX 1 X - Drive":PRINT#2,"J
     X 2 X - CAT":PRINT#2,"JX 3 X - loes
     chen"                                [D17C]
1240 PRINT#2,"JX 4 X - umbenennen";:PRIN
     T#2,"JX 5 X - USER":PRINT#2,"JJX 9
     X - Ende"                            [4DA6]
1250 PRINT#1,"LJ";USING "Laufwerk: \ \<3
     9>User: ##";drive$;user              [FCF2]
1260 CLS:CAT                              [E6BA]
1270 jn=ASC(INKEY$+CHR$(0))-48            [7722]
1280 IF jn=9 THEN 3760 ELSE IF jn<1 OR j
     n>5 THEN 1270                        [FCD2]
1290 IF jn=1 THEN 1380                    [1C38]
1300 IF jn=2 THEN 1260                    [1924]
1310 IF jn=3 THEN 1410                    [F922]
1320 IF jn=4 THEN 1460                    [D330]
1330 ' *****************

          * USER wechseln *
                              *************
     ****                                 [129C]
1340 IF jn=5 THEN l=2:t=2:PRINT"JGUsernu
     mmer: ";:GOSUB 3400                  [7D9A]
1350 IF ret THEN 1260 ELSE user=VAL(in$)
     :IF user>15 THEN user=15 ELSE IF us
     er<0 THEN user=0                     [3F54]
1360 |USER,user:GOTO 1250                 [94D6]
1370 ' ******************

          * DRIVE wechseln *
                              *************
     *****                                [8242]
1380 IF drive$="A" THEN |B:drive$="B":GO
     TO 1250                              [B8BC]
1390 |A:drive$="A":GOTO 1250              [DC7E]
1400 ' *****************

          * File loeschen *
                              *************
     ****                                 [6D0A]
1410 PRINT"JName des zu loeschenden File
     s: ";:l=12:GOSUB 3400:IF ret THEN 1
     260                                  [ACA0]
1420 PRINT"JSind Sie SICHER? (J/N)":jn$=
     ""                                   [473A]
1430 WHILE jn$<>"J" AND jn$<>"N":jn$=UPP
     ER$(INKEY$):WEND:IF jn$="N" THEN 12
     60                                   [0018]
1440 |ERA,@in$:GOTO 1260                  [EB00]
1450 ' *******************

          * File umbenennen *
                              *************
     ******                               [EC18]
1460 PRINT"JName des alten Files: ";:l=1
     2:GOSUB 3400:IF ret THEN 1260        [B9A4]
1470 o$=in$                               [076E]
1480 PRINT"JName des neuen Files: ";:l=1
     2:GOSUB 3400:IF ret THEN 1260        [EBB6]
1490 n$=in$                               [3670]
1500 |REN,@n$,@o$                         [9088]
1510 GOTO 1260                            [9112]
1520 ' ****************

          * Druckermenue *
                              *************
     ***                                  [83DA]
1530 IF dat$="" THEN PRINT"G":GOTO 3760   [0040]
1540 dinit=0:CLS:PRINT#1,"************ D
     ruckerinit *************";           [E55E]
1550 PRINT"OAJJFeldnamen:":FOR n=1 TO fe
     ld                                   [6F7E]
1560 LOCATE 1,6:PRINT"X "RIGHT$(feld$(n)
     ,LEN(feld$(n))-8)" X drucken? ";     [DCD8]
1570 jn$="":WHILE jn$<>"J" AND jn$<>"N":
     jn$=UPPER$(INKEY$):WEND:PRINT jn$:d
     r$(n)=jn$                            [BBD6]
1580 LOCATE 1,6:PRINT SPACE$(40):NEXT     [EEFC]
1590 CLS:PRINT"OBFeldinhalt:":PEN 1       [7A20]
1600 FOR n=1 TO feld:LOCATE 1,6:IF n<>fe
     ld THEN PRINT#2,RIGHT$(feld$(n+1),L
     EN(feld$(n+1))-8)                    [A77C]
1610 PRINT"X "RIGHT$(feld$(n),LEN(feld$(
     n))-8)" X : ";                       [A5A4]
1620 GOSUB 1680:IF ret THEN 3760 ELSE dr
     $(n)=dr$(n)+in$                      [2912]
1630 LOCATE 1,6:PRINT SPACE$(40):NEXT     [49F4]
1640 CLS:PRINT"OAJJJZeilenvorschub: ";:l
     =4:t=2                               [BB44]
1650 GOSUB 3400:IF ret THEN 3760 ELSE vo
     r=VAL(in$)                           [40E4]
1660 dinit=1:GOTO 3760                    [E2AC]
1670 ' ****************

          * Inkeyroutine *
                              *************
     ***                                  [F31E]
1680 ret=0:y=VPOS(#0):x=POS(#0):in$="":l
     =0                                   [6BA2]
1690 LOCATE x,y:PRINT STRING$(5,95)       [81CE]
1700 a$="":WHILE a$="":a$=INKEY$:WEND     [489E]
1710 IF a$=CHR$(224) THEN ret=1:RETURN    [120C]
1720 IF a$=CHR$(13) THEN RETURN           [14C0]
1730 IF a$=CHR$(127) AND l>0 THEN l=l-1:
     LOCATE x+l,y:PRINT"_":in$=LEFT$(in$
     ,LEN(in$)-1):GOTO 1700               [C5C8]
1740 IF l=5 THEN SOUND 1,0,1,7,,,10:GOTO
      1700                                [E93C]
1750 IF a$=CHR$(243) THEN LOCATE x+l,y:P
     RINT CHR$(243):in$=in$+CHR$(243):l=
     l+1:GOTO 1700                        [C03E]
1760 IF a$=CHR$(241) THEN LOCATE x+l,y:P
     RINT CHR$(241):in$=in$+CHR$(241):l=
     l+1:GOTO 1700                        [3F34]
1770 SOUND 1,50,3,7,,,1:GOTO 1700         [5958]
1780 ' *****************

          * Daten drucken *
                              *************
     ****                                 [382E]
1790 PRINT#1,"*********** Daten drucken
     ************";                       [DDF6]
1800 CLS                                  [2D96]
1810 IF dat$="" OR er=1 THEN 3760         [7DB8]
1820 IF dinit=0 THEN CLS:PRINT"JJJJJJJOA
     GBitte erst X Drucker einstellen X
     !!":FOR n=1 TO 2000:NEXT:GOTO 3760   [8CC2]
1830 CLS:PRINT"Wollen Sie:J"              [EF9E]
```

**Listing 2. Mit »Datafine« haben Sie Ihre Daten fest im Griff (Fortsetzung)**

```
1840 PRINT"OBX 1 XOA - alle Daten drucke
     n?J"                                    [9DD8]
1850 PRINT"OBX 2 XOA - nach Suchbegriff
     drucken?J"                              [EACC]
1860 PRINT"OBX 3 XOA - best. Datensatz d
     rucken?J"                               [57E0]
1870 PRINT"NAOCX 9 XOAND - Beenden?"         [5514]
1880 jn=ASC(INKEY$+CHR$(0)):jn=jn-48         [227E]
1890 IF jn=9 THEN 3760 ELSE IF jn<1 OR j
     n>3 THEN 1880                           [F7EA]
1900 ON jn GOTO 1910,1990,2180               [AD72]
1910 '                                       [BE24]
1920 MODE 2:GOSUB 3500                       [8E0A]
1930 OPENIN ""+dat$                          [4B44]
1940 FOR num=0 TO daz-1:a$=SPACE$(laenge
     ):po=num*laenge:|R,@po,@a$              [AEF6]
1950 GOSUB 3390:LOCATE 1,1:GOSUB 3250        [2C44]
1960 GOSUB 3170                              [D3B6]
1970 IF INKEY$=CHR$(224) THEN num=daz        [134C]
1980 NEXT num:GOTO 3760                      [4B08]
1990 '                                       [9634]
2000 CLS:PRINT"JJSuchbegriffG:":l=35         [BCAE]
2010 GOSUB 3400:IF ret THEN 3760 ELSE su
     $=in$                                   [C4D4]
2020 PRINT"JJSuchen von Datensatz ";:l=4
     :t=2:GOSUB 3400                         [3140]
2030 IF ret THEN 3760 ELSE sua=VAL(in$):
     IF sua<0 THEN sua=0 ELSE IF sua>daz
     -1 THEN sua=daz-1                       [459A]
2040 PRINT"JJSuchen bis Datensatz ";:l=4
     :t=2:GOSUB 3400                         [AB1A]
2050 IF ret THEN 3760 ELSE sue=VAL(in$):
     IF sue<sua THEN sue=sua ELSE IF sue
     >daz-1 THEN sue=daz-1                   [052A]
2060 MODE 2:GOSUB 3500:OPENIN ""+dat$        [8FE0]
2070 FOR num=sua TO sue:a$=SPACE$(laenge
     ):po=num*laenge:|R,@po,@a$              [F07E]
2080 LOCATE #1,1,1:PRINT#1,USING "Datens
     atz ####";num                           [90CC]
2090 IF INSTR(a$,su$) THEN 2120              [6AD8]
2100 NEXT num                                [4824]
2110 PRINT"LJJGEnde der Suche":FOR n=1 T
     O 2500:NEXT:GOTO 3760                   [9FDC]
2120 CLS:GOSUB 3390:GOSUB 3250:PRINT#1,"
     LDrucken? (J/N)":jn$=""                 [EA4A]
2130 WHILE jn$<>"J" AND jn$<>"N":jn$=UPP
     ER$(INKEY$):WEND:IF jn$="N" THEN 21
     50                                      [0B12]
2140 GOSUB 3170                              [5DA4]
2150 PRINT#1,"LJGWeitersuchen ?<2>(J/N)G
     "                                       [DE14]
2160 jn$="":WHILE jn$<>"J" AND jn$<>"N":
     jn$=UPPER$(INKEY$):WEND                 [28B0]
2170 IF jn$="J" THEN CLS #1:GOTO 2100 EL
     SE 3760                                 [BF68]
2180 MODE 2:GOSUB 3500:OPENIN ""+dat$        [CAE6]
2190 CLS:PRINT"JDatensatznummer: ";:l=4:
     t=2:GOSUB 3400:IF ret THEN 3760         [1E04]
2200 CLS:num=VAL(in$):IF num<0 THEN num=
     0 ELSE IF num>daz-1 THEN num=daz-1      [5ED2]
2210 a$=SPACE$(laenge):po=num*laenge:|R,
     @po,@a$                                 [6DE8]
2220 GOSUB 3390:GOSUB 3250:PRINT#1,"LJDr
     ucken? (J/N)"                           [33BA]
2230 jn$="":WHILE jn$<>"J" AND jn$<>"N":
     jn$=UPPER$(INKEY$):WEND                 [21AC]
2240 IF jn$="N" THEN 2260                    [D436]
2250 GOSUB 3170                              [5EA8]
2260 PRINT#1,"LJWeiteren Datensatz druck
     en? (J/N)"                              [92CA]
2270 jn$="":WHILE jn$<>"J" AND jn$<>"N":
     jn$=UPPER$(INKEY$):WEND:IF jn$="N"
     THEN 3760 ELSE 2190                     [4304]
2280 ' *******************

               * Daten sortieren *

                               *************
     ******                                  [F754]
2290 IF dat$="" OR er=1 THEN GOSUB 3290:
     IF dat$="" OR er=1 THEN 3760            [BE52]
2300 CLS:CLS #1:PRINT#1,"********** Date
     n sortieren ***********";               [026C]
2310 PRINT"OANach welchem Feld sortieren
     :"                                      [562C]
2320 FOR n=1 TO feld:feld$=RIGHT$(feld$(
     n),LEN(feld$(n))-8):PRINT"OBX ";n;"
     XOA - ";feld$:NEXT                      [B6A4]
2330 PRINT"JBitte Feldnummer eingeben: "
     ;:l=2:t=2:GOSUB 3400:IF ret THEN 37
     60 ELSE sort=VAL(in$):IF sort<1 OR
     sort>feld THEN PRINT"KKK":GOTO 2330
                                             [FF36]
2340 GOSUB 2350:GOTO 3760                    [A16C]
2350 OPENIN ""+dat$                          [E93E]
2360 FOR i=1 TO daz-1:fl=0                   [A084]
2370 FOR j=daz-1 TO i STEP -1                [6ABA]
2380 a$=SPACE$(laenge):po=(j-1)*laenge:|
     R,@po,@a$:x1$=a$:GOSUB 3390             [AB6E]
2390 a1$=i$(sort):a$=SPACE$(laenge):po=j
     *laenge:|R,@po,@a$:x2$=a$:GOSUB 339
     0:a2$=i$(sort)                          [3262]
2400 IF a1$>a2$ THEN po=(j-1)*laenge:|W,
     @po,@x2$:po=j*laenge:|W,@po,@x1$:fl
     =1                                      [9DDA]
2410 IF INKEY$=CHR$(224) THEN j=i:fl=0       [8DB2]
2420 NEXT j:IF fl=0 THEN RETURN              [2B50]
2430 NEXT i:RETURN                           [6C96]
2440 ' *************************

               * Neue Datei einrichten
     *
                               *************
     ************                            [F82C]
2450 CLS:PEN 3:PRINT"Datei einrichten:"      [2B64]
2460 PRINT"JJDAX 1 X - neue Maske defini
     eren":PRINT"JX 2 X - Maske aus Date
     i laden?"                               [459C]
2470 jn=0:WHILE jn<1 OR jn>2:jn=ASC(INKE
     Y$+CHR$(0))-48:WEND:IF jn=1 THEN ma
     s=0:GOTO 2600                           [FDC6]
2480 CLS:PEN 1:PRINT"Maske aus Datei lad
     en:"                                    [5B10]
2490 PRINT:PEN 2:PRINT"Name der Datei:";
     :l=8:PEN 1:GOSUB 3400                   [60F0]
2500 IF ret THEN 3760 ELSE dat$=UPPER$(i
     n$):er=0:GOSUB 3670                     [508E]
2510 IF er=1 THEN PRINT"JJGDatei nicht v
     orhanden!!":FOR n=1 TO 2000:NEXT:GO
     TO 2490                                 [2930]
2520 CLS:PRINT"JMaskendatei wird geladen
     !":GOSUB 3340                           [F9A2]
2530 MODE 2:num=0:GOSUB 3510                 [88F6]
2540 a$=STRING$(laenge,95):GOSUB 3390:GO
     SUB 3250                                [0098]
2550 PRINT#1,"GMaske in Ordnung (J/N)?":
     jn$=""                                  [6EAE]
2560 WHILE jn$<>"J" AND jn$<>"N":jn$=UPP
     ER$(INKEY$):WEND                        [C14A]
2570 MODE 1:GOSUB 3610                       [6810]
2580 IF jn$="N" THEN 2450                    [E046]
2590 mas=1                                   [903E]
2600 PEN 2:PRINT"JName der neuen Datei:"
     :PEN 1:l=8:GOSUB 3400                   [5B68]
2610 IF ret THEN 3760 ELSE dat$=UPPER$(i
     n$)                                     [4542]
2620 PEN 2:PRINT"JGroesse der Datei: ";:
     PEN 1:l=4:t=2:GOSUB 3400                [908C]
2630 IF ret THEN 3760 ELSE anz=VAL(in$):
     IF anz<1 OR anz>1000 THEN PRINT"G":
     CLS:GOTO 2620                           [4594]
2640 GOSUB 2690:ERASE dr$:DIM dr$(feld):
     CLS:PRINT"JJJJ<3>Bitte warten!"         [0B68]
2650 OPENOUT ""+dat$+".ind":PRINT#9,anz:
     PRINT#9,0:CLOSEOUT                      [6134]
2660 OPENOUT ""+dat$:FOR n=1 TO anz:PRIN
     T#9,SPACE$(laenge):LOCATE 16,8          [AF8C]
2670 PRINT USING "####";n:NEXT:PRINT#9,S
     PACE$(128):PRINT#9,SPACE$(128):CLOS
     EOUT:CLS:RETURN                         [3950]
2680 ' ***************************

               * Eingabemaske definier
     en *
                               *************
     **************                          [4EB8]
2690 IF mas=1 THEN 3080 ELSE MODE 2:GOSU
     B 3500                                  [4616]
2700 x=1:y=1:LOCATE 1,1:PRINT"X X":c$="X
     X"                                      [A1FA]
2710 PRINT#1,"Cursortasten: Cursor steue
     rn":PRINT#1,"CTRL+E: Eingabe beende
     n - Definition"                         [B1C4]
2720 ' *****************

               * Maske aufbauen *

                               *************
     *****                                   [CA0C]
2730 jn=ASC(INKEY$+CHR$(0)):IF jn=0 THEN
      2730 ELSE IF jn=5 THEN LOCATE x,y:
     PRINT MID$(c$,2,1):GOTO 2870            [DB80]
2740 IF jn<240 OR jn>243 THEN 2820           [C022]
2750 x1=x:y1=y:c1$=MID$(c$,2,1)              [5E4C]
2760 x=x+(1 AND jn=243)-(1 AND jn=242):I
     F x>58 THEN x=1:y=y+(1 AND y<18) EL
     SE IF x<1 THEN x=58:y=y-(1 AND y>1)
                                             [E384]
2770 y=y+(1 AND jn=241)-(1 AND jn=240):I
     F y>18 THEN y=1 ELSE IF y<1 THEN y=
     18                                      [1AC6]
2780 IF x=x1 AND y=y1 THEN 2730 ELSE LOC
     ATE x,y:CALL 40000:c$="X"+CHR$(PEEK
     (40010))+"X"                            [447A]
2790 LOCATE x1,y1:PRINT c1$;:LOCATE x,y:
     PRINT c$;                               [F5F6]
```

```
2800 LOCATE #2,1,1:PRINT#2,USING "x:## y
     :##";x;y                                [51F6]
2810 GOTO 2730                               [9E20]
2820 IF jn=127 THEN c$="X X":jn=242:GOTO
      2750                                   [5DB2]
2830 IF jn=13 THEN LOCATE x,y:PRINT" ":c
     $="X X":x=58:jn=243:GOTO 2750           [CCB6]
2840 c$="X"+CHR$(jn)+"X":jn=243:GOTO 275
     0                                       [8366]
2850 GOTO 2730                               [7A28]
2860 ' ********************

                   * Maskendefinition *

                              *************
     *******                                 [18C6]
2870 ERASE feld$:DIM feld$(20):glang=255
     :feld=0:er=0:PRINT#1,"LJBitte warte
     n. Maske wird definiert!":CLS #2        [105A]
2880 FOR y=1 TO 18:FOR x=1 TO 58             [EC46]
2890 LOCATE x,y:CALL 40000:c=PEEK(40010)
                                             [F32E]
2900 IF c=58 AND feld>=20 THEN er=1:x=58
     :y=18:PRINT#1,"LJGMehr als 20 Felde
     r definiert!!":GOTO 2930                [0B40]
2910 IF c=58 THEN feld=feld+1:GOSUB 2960
     :GOSUB 3000:IF er=1 THEN PRINT#1,"L
     JGFalsches Format!!":x=58:y=18:GOTO
      2930                                   [F986]
2920 glang=glang-lang:LOCATE #1,1,1:PRIN
     T#1,glang:lang=0:IF glang<0 THEN PR
     INT#1,"LJGLaenge der Felder zu gros
     s (>255)!!"::x=58:y=18:er=1:GOTO 29
     30                                      [77A2]
2930 NEXT x,y:IF er=1 THEN x=1:y=1:LOCAT
     E x,y:CALL 40000:c$="X"+CHR$(PEEK(4
     0010))+"X":LOCATE x,y:PRINT c$:CLS
     #2:GOTO 2750                            [0FFE]
2940 GOTO 3080                               [AF26]
2950 ' **** Feld erfassen ****               [4872]
2960 FOR x1=x-1 TO 1 STEP -1                 [4A44]
2970 LOCATE x1,y:CALL 40000:c1=PEEK(4001
     0):IF c1=32 THEN x1=1:GOTO 2980 ELS
     E feld$(feld)=CHR$(c1)+feld$(feld)      [2B20]
2980 NEXT:PRINT#2,feld$(feld):RETURN         [227E]
2990 ' **** Laenge und Typ des Feldes **
     **                                      [D854]
3000 lang=0:typ=0:FOR x1=x+1 TO 58           [D782]
3010 LOCATE x1,y:CALL 40000:c1=PEEK(4001
     0):IF x1=x+1 AND c1<>32 THEN er=1:x
     1=58:GOTO 3050                          [9C5E]
3020 IF c1=35 THEN lang=lang+1:IF typ=0
     THEN typ=2 ELSE IF typ=1 THEN er=1:
     x1=58:GOTO 3050                         [480C]
3030 IF c1=36 THEN lang=lang+1:IF typ=0
     THEN typ=1 ELSE IF typ=2 THEN er=1:
     x1=58:GOTO 3050                         [DC10]
3040 IF c1=32 AND x1>x+1 THEN x1=58:GOTO
      3050                                   [2A6E]
3050 NEXT x1:IF er=1 THEN 3070               [95AE]
3060 feld$(feld)=HEX$(lang,2)+HEX$(typ,2
     )+HEX$(x-LEN(feld$(feld)),2)+HEX$(y
     ,2)+feld$(feld)                         [2100]
3070 RETURN                                  [8B94]
3080 MODE 1:PRINT"Folgende Felder sind d
     efiniert:J"                             [96DC]
3090 FOR n=1 TO feld:PRINT feld$(n):NEXT
                                             [7256]
3100 PRINT"JMaske wird gespeichert!"         [25B4]
3110 laenge=0:FOR n=1 TO feld:laenge=lae
     nge+VAL("&"+LEFT$(feld$(n),2)):NEXT
                                             [D722]
3120 OPENOUT ""+dat$+".msk"                  [55C8]
3130 PRINT#9,feld:PRINT#9,laenge:FOR n=1
      TO feld:PRINT#9,feld$(n):NEXT          [EA66]
3140 CLOSEOUT                                [48AC]
3150 ERASE i$:DIM i$(feld):RETURN            [839A]
3160 ' *****************

                   * Daten drucken *

                              *************
     ****                                    [3022]
3170 FOR n=1 TO feld:feld$=RIGHT$(feld$(
     n),LEN(feld$(n))-8)                     [5B64]
3180 IF LEFT$(dr$(n),1)="J" AND LEN(dr$(
     n))>1 THEN PRINT#8,feld$;": ";          [F6BA]
3190 d$=RIGHT$(dr$(n),LEN(dr$(n))-1)         [DEA2]
3200 IF d$="" THEN 3220 ELSE PRINT#8,i$(
     n);" ";:FOR nn=1 TO LEN(d$):IF MID$
     (d$,nn,1)=CHR$(241) THEN PRINT#8        [F57C]
3210 NEXT                                    [E14A]
3220 NEXT:FOR n=1 TO vor:PRINT#8:NEXT        [F730]
3230 RETURN                                  [8D90]
3240 ' **********************

                   * Datensatz anzeigen *

                              *************
     *********                               [EF36]
3250 FOR n=1 TO feld:x=VAL("&"+MID$(feld
     $(n),5,2)):y=VAL("&"+MID$(feld$(n),
     7,2))                                   [BDAC]
3260 feld$=RIGHT$(feld$(n),LEN(feld$(n))
     -8):LOCATE x,y:PRINT feld$;": ";i$(
     n):NEXT                                 [308A]
3270 LOCATE #2,1,22:PRINT#2,USING "Ds.:
     ####";num:RETURN                        [7C50]
3280 ' ******************

                   * Datei eroeffnen *
                              *************
     ******                                  [2306]
3290 CLS:PEN 1:PRINT"Datei noch nicht ge
     oeffnet!!":PRINT:PEN 2                  [25F2]
3300 PRINT"Dateiname: ";:PEN 1:l=8:GOSUB
      3400                                   [50FC]
3310 IF ret THEN 3370 ELSE dat$=in$:er=0
     :GOSUB 3660                             [E884]
3320 IF er=1 THEN PEN 1:PRINT"JJGDatei n
     icht vorhanden!!":FOR n=1 TO 2000:N
     EXT:GOTO 3290                           [170A]
3330 CLS:OPENIN ""+dat$+".ind":INPUT #9,
     anz:INPUT #9,daz:CLOSEIN                [A68A]
3340 ERASE feld$,i$,dr$:OPENIN ""+dat$+"
     .msk":INPUT#9,feld:INPUT#9,laenge       [FA34]
3350 DIM feld$(feld),i$(feld),dr$(feld)      [BA8E]
3360 FOR n=1 TO feld:LINE INPUT#9,feld$(
     n):NEXT n:CLOSEIN                       [9F66]
3370 RETURN                                  [AA9A]
3380 ' ******************

                   * Stringzuweisung *

                              *************
     ******                                  [635A]
3390 FOR n=1 TO feld:l=VAL("&"+LEFT$(fel
     d$(n),2)):i$(n)=LEFT$(a$,l):a$=RIGH
     T$(a$,LEN(a$)-l):NEXT n:RETURN          [5D0E]
3400 ' *** Inputroutine ***

                   *Uebergabeparameter*

                              * l=Laenge  t
     =typ **                                 [0F30]
3410 ret=0:y=VPOS(#0):x=POS(#0):ll=0:in$
     =""                                     [B96C]
3420 LOCATE x,y:PRINT STRING$(1,"_")         [AA98]
3430 a$="":WHILE a$="":a$=INKEY$:WEND:a=
     ASC(a$):IF a=224 THEN ret=1:t=1:RET
     URN ELSE IF a=13 THEN t=1:RETURN        [69B0]
3440 IF a$=CHR$(127) AND ll>0 THEN 3490
     ELSE IF a$=CHR$(127) AND ll=0 THEN
     3430                                    [666E]
3450 IF t=2 AND (a<48 OR a>57) THEN SOUN
     D 2,500,3,7,,,10:GOTO 3430              [3CCC]
3460 IF ll=l THEN SOUND 1,2500,3,7:GOTO
     3430                                    [D4F0]
3470 in$=in$+a$:ll=ll+1:LOCATE x+ll-1,y:
     PRINT a$                                [E7A0]
3480 GOTO 3430                               [DB24]
3490 in$=LEFT$(in$,ll-1):ll=ll-1:LOCATE
     x+ll,y:PRINT"_H":SOUND 1,1000,1,7:G
     OTO 3430                                [FDD8]
3500 ' ************************

                   * Bildschirmmaske MODE
     2 *
                              *************
     *************                           [1CE8]
3510 ol$=CHR$(150):ore$=CHR$(156):ul$=CH
     R$(147):ur$=CHR$(153):wa$=CHR$(154)
     :se$=CHR$(149)                          [1DEA]
3520 LOCATE 1,1:PRINT ol$;STRING$(60,wa$
     );ore$;ol$;STRING$(16,wa$);ore$         [1BCA]
3530 FOR n=2 TO 19:LOCATE 1,n:PRINT se$:
     LOCATE 62,n:PRINT se$;se$:LOCATE 80
     ,n:PRINT se$:NEXT                       [8CF4]
3540 LOCATE 1,20:PRINT ul$;STRING$(60,wa
     $);ur$;se$;TAB(80);se$                  [7206]
3550 LOCATE 1,21:PRINT ol$;STRING$(60,wa
     $);ore$;se$;TAB(80);se$                 [BABC]
3560 FOR n=22 TO 24:LOCATE 1,n:PRINT se$
     :LOCATE 62,n:PRINT se$;se$:LOCATE 8
     0,n:PRINT se$:NEXT                      [3D56]
3570 LOCATE 1,25:PRINT ul$;STRING$(60,wa
     $);ur$;ul$;STRING$(16,wa$);ur$;         [B852]
3580 WINDOW 2,61,2,19:WINDOW #1,2,61,22,
     24:WINDOW #2,64,79,2,24                 [C7C6]
3590 PEN #1,1:PEN #2,1:RETURN                [5C5C]
3600 ' ************************
                   * Bildschirmmaske MODE
     1*
                              *************
     ************                            [5F80]
```

**Listing 2. Mit »Datafine« haben Sie Ihre Daten fest im Griff (Fortsetzung)**

```
3610 a1$=CHR$(194)+STRING$(38,154)+CHR$(
     195):a2$=CHR$(193)+STRING$(38,154)+
     CHR$(192):a3$=CHR$(149)               [6E1C]
3620 PEN#3,3:LOCATE #3,1,1:PRINT#3,a1$;a
     3$;TAB(40)a3$;a2$;:PRINT#3,a1$;       [D1FA]
3630 FOR n=5 TO 20:LOCATE #3,1,n:PRINT#3
     ,a3$;:LOCATE #3,40,n:PRINT #3,a3$;:
     NEXT                                  [8334]
3640 PRINT#3,a2$;a1$;a3$TAB(40);a3$;a3$T
     AB(40)a3$;a2$;                        [A6F4]
3650 WINDOW 2,39,5,20:WINDOW #1,2,39,2,2
     :WINDOW #2,2,39,23,24:FOR n=0 TO 2:
     CLS #n:NEXT:PEN #1,2:RETURN           [DEEE]
3660 ' ********************

              * Directory lesen *

                          **************
     ******                                [3964]
3670 ERASE a$:DIM a$(65):dat1$=LEFT$(UPP
     ER$(dat$)+SPACE$(8),8)+".<3>"         [3A44]
3680 a=PEEK(&BB5A):POKE &BB5A,&C9:CAT:PO
     KE &BB5A,a                            [D9C8]
3690 anz=PEEK(&A912):a=PEEK(&A79C)*256+P
     EEK(&A79B)+1                          [426C]
3700 FOR an=0 TO anz:FOR n=a TO a+10:x=P
     EEK(n):a$(an)=a$(an)+CHR$(x):NEXT:a
     =a+14:NEXT                            [0F74]
3710 FOR n=0 TO (anz AND anz<64)+(63 AND
      anz>=64):IF ASC(LEFT$(a$(n),1))=0
     THEN a=n:n=anz:GOTO 3730              [E074]
3720 a$(n)=LEFT$(a$(n),8)+"."+RIGHT$(a$(
     n),3)                                 [3426]
3730 NEXT:anz=a                            [A29A]
3740 FOR n=0 TO anz:IF dat1$=a$(n) THEN
     3750 ELSE NEXT:er=1                   [F0B2]
3750 RETURN                                [8A9E]
3760 ' *********

                  * Menue *

                         *********         [D6F2]
3770 CLOSEIN                               [60FC]
3780 MODE 1:GOSUB 3600:PRINT#1,"********
     ***** DATAFINE CPC ************";     [9FBA]
3790 CLS:RESTORE 3970:FOR n=1 TO 8:READ
     a$:LOCATE 8,n*2-1                     [0CEE]
3800 IF n<8 THEN PEN 1:PRINT"X"n"X - ";
     ELSE PAPER 1:PEN 3:PRINT"X 9 X";:PA
     PER 0:PEN 1:PRINT" - ";               [B1D0]
3810 PEN 2:PRINT a$:PRINT:NEXT             [9F9A]
3820 ' ******************************

              * Tastaturabfrage Zahle
     n 1-9 *
                          *************
     *****************                     [02F2]
3830 PRINT#2,TAB(9);CHR$(164);" 1986 by
     ms-Software "CHR$(255)                [33FA]
3840 IF NOT(dat$="") AND er=0 THEN PRINT
     #2,TAB(9);"OBGeoeffnete Datei: ";da
     t$;"OA";                              [697C]
3850 a$=INKEY$:IF a$="" THEN 3850          [0C08]
3860 a=ASC(a$+CHR$(0))-48:IF a<1 OR a>9
     OR a=8 THEN SOUND 1,500,2,5:GOTO 38
     50                                    [AAAA]
3870 IF a=9 THEN a=8                       [AA7A]
3880 LOCATE 5,(a-1)*2+1:PEN 3:PRINT STRI
     NG$(3,243)                            [001E]
3890 ' **********************

              * Sicherheitsabfrage *

                          *************
     *********                             [9F94]
3900 PRINT#2,"GJ<8>Sind Sie sicher? (J/N
     )":a$=""                              [DEE6]
3910 WHILE a$<>"J" AND a$<>"N":a$=UPPER$
     (INKEY$):WEND:CLS #2:IF a$="J" THEN
      3930                                 [E690]
3920 LOCATE 5,(a-1)*2+1:PRINT"<3>":GOTO
     3850                                  [697A]
3930 ON a GOTO 210,410,1170,1220,1530,17
     90,2290,3940                          [EA4E]
3940 MODE 2:BORDER 1:INK 0,1:INK 1,24:PE
     N 1:PAPER 0:END                       [8EF2]
3950 GOTO 3950                             [8C36]
3960 ' *********************

              * Datas fuer Menue *

                          *************
     *******                               [0A6C]
3970 DATA Daten eingeben,Datei pflegen,D
     atei wechseln,Diskmenue,Drucker ein
     stellen,Daten drucken,Daten sortier
     en,GENDE                              [BB78]
```

**Listing 2. Mit »Datafine« haben Sie Ihre Daten fest im Griff (Schluß)**

# Wem die Stunde schlägt

**Zeit ist Geld, sagt ein kluges Sprichwort - nicht ganz zu unrecht. Der Computer hilft Ihnen, Termine perfekt zu planen.**

Stehen Sie auf Kriegsfuß mit Ihrer Terminplanung? Wenn Sie einen Schneider CPC Ihr eigen nennen und dieses Listing eingeben, verwaltet der Computer Ihre Termine automatisch. Das Programm »Termin« arbeitet menügesteuert, so daß es sofort und ohne Probleme nutzbar ist. Zur Bearbeitung steht immer ein Monat des aktuellen Jahres bereit. Damit die Wochentage dem jeweiligen Datum korrekt zugeordnet sind, enthält Termin einen immerwährenden Kalender, der – für eine aktuelle Terminplanung etwas überflüssig – ab dem Jahr 1513 zählt. Im Hauptmenü wählen Sie mit den Cursor-Tasten aus sechs Punkten einen aus und aktivieren die Funktion mit der Taste <COPY>. Der Reihenfolge auf dem Bildschirm nach, arbeiten die Menüpunkte wie folgt:

**Eingeben**

Der Monitor zeigt die Daten der ersten Hälfte des aktuellen Monats. Neben Datum und Wochentag haben Sie rechts Platz für tägliche Eintragungen mit einer Länge von jeweils bis zu 30 Zeichen. Zur Eingabe bewegen Sie die weiß unterlegte Zeile mit Hilfe der Cursor-Tasten auf den gewünschten Tag und drücken dort <COPY>. Dann machen Sie Ihren Eintrag und beenden mit der Taste <ENTER>. Auf die zweite Bildschirm-Seite (zweite Monatshälfte) gelangen Sie durch gleichzeitigen Druck der Tasten <CTRL+6> (<F6> beim CPC 6128) auf dem Zehnerblock. <CTRL+5> führt zurück zur ersten Seite und <CTRL+9> ins Menü. Einen Eintrag löscht eine Neueingabe oder <CTRL+8>.

**Drucken**

Haben Sie einen Drucker angeschlossen, lassen sich die Termine jeden Monat schwarz auf weiß drucken.

**Zeitraum**

Mit den Cursor-Tasten wählen Sie einen neuen Monat des aktuellen Jahres zur Bearbeitung.

**Laden**

Hier laden Sie die gespeicherten Daten eines Jahres-Terminkalenders.

**Speichern**

Bevor Sie neue Daten laden oder die Arbeit beenden, müssen Sie geänderte oder eingegebene Daten speichern, um Datenverlusten vorzubeugen. Der Name der Datei ergibt sich automatisch aus der anfangs eingegebenen Jahreszahl.

**Directory**

Arbeiten Sie mit einem Diskettenlaufwerk, haben Sie Zugriff auf das Inhaltsverzeichnis. Wenn Sie den CPC 464 nur mit dem eingebauten Kassettenrecorder benutzen, ignorieren Sie diesen Menüpunkt.

Verlassen Sie das Programm jemals versehentlich mit <ESC>, kommen Sie ohne jeden Datenverlust mit »GOTO 110« wieder in das Menü zurück.

Natürlich läßt sich mit dem immerwährenden Kalender auch jedes beliebige Datum bestimmen. Den Wochentag Ihres Geburtstags erfahren Sie beispielsweise, indem Sie beim Programmstart Ihr Geburtsjahr und dann unter »Zeitraum« den betreffenden Monat wählen. Gehen Sie dann in den Eingabemodus, erhalten Sie die gewünschte Information. (Patrick Schuster/ja)

| Steckbrief | |
|---|---|
| Programm: | Terminkalender |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

```
10 '###################################       [4E56]
20 '##                               ##       [D1AA]
30 '## ----> Terminkalender <----    ##       [A7DA]
40 '##                               ##       [91AE]
50 '##      By Patrick Schuster      ##       [3324]
60 '##          12.08.1986           ##       [C130]
70 '##                               ##       [75B4]
80 '###################################       [8264]
90 '                                          [8560]
100 DIM menue$(12),a(12),b(12),day$(7),d
    ay(7),eintr$(12,32),monla(12),mona(1
    2)                                    [FAC2]
110 ON ERROR GOTO 2470                    [607E]
120 rag=0:GOSUB 130:GOTO 180              [ACE8]
130 mona$=" Januar<2>":ma=1:m$=" *Menue*
     ":i$=CHR$(24)                        [65A2]
140 invon$=CHR$(22)+CHR$(1):invoff$=CHR$
    (22)+CHR$(0)                          [9854]
150 RESTORE 1890:FOR t=1 TO 7:READ day$(
    t):day(t)=t:NEXT t                    [F942]
160 RETURN                                [A62E]
170 :                                     [CDE4]
180 PEN 1:INK 0,0:INK 2,15:INK 3,6:INK 1
    ,26:BORDER 0:PAPER 0                  [5EA2]
190 CLS:MODE 1:GOSUB 220:LOCATE 1,23:GOS
    UB 220                                [73F8]
200 GOTO 250                              [EB44]
210 :                                     [CBDA]
220 PRINT CHR$(150);STRING$(38,154);CHR$
    (156);CHR$(149);STRING$(38,207);CHR$
    (149);CHR$(147);STRING$(38,154);CHR$
    (153);                                [D7F2]
230 RETURN                                [A92A]
240 :                                     [CAE0]
250 LOCATE 13,2:PRINT "Termin-Kalender";
                                          [9AF2]
260 WINDOW#0,1,40,4,22:WINDOW #2,9,39,5,
    21                                    [E268]
270 LOCATE 1,5:PRINT "Geben sie das Jahr
     ein ";:LINE INPUT"";jahr$:jahr=VAL(
    jahr$)                                [E26E]
280 IF jahr<1512 OR jahr>2222 THEN 270    [814E]
290 GOSUB 2210:CLS                        [8B78]
300 WINDOW SWAP 1,0:LOCATE 33,24:PRINT i
    nvoff$;jahr;:WINDOW SWAP 1,0          [87CA]
310 GOSUB 2190                            [2040]
320 '---------------                      [B1A4]
330 '- Hauptmenue -                       [D226]
340 '---------------                      [63A8]
350 WINDOW SWAP 1,0:LOCATE 18,24:PRINT "
    Hauptmenue";:WINDOW SWAP 1,0          [1682]
360 RESTORE 1850:FOR t=1 TO 6:READ menue
    $(t):a$=menue$(t):READ a(t),b(t):x=a
    (t):y=b(t):GOSUB 1780:NEXT t          [90A2]
370 GOSUB 2190                            [144C]
380 flag=0                                [3884]
390 wahl=1:px=2:py=3                      [4AF4]
400 PEN 1:PRINT invoff$;:LOCATE a(wahl),
    b(wahl):PRINT CHR$(24);menue$(wahl);
    CHR$(24);invon$;                      [C6FA]
410 GOTO 1950                             [13BA]
420 :                                     [E0E0]
430 IF in=9 THEN 470                      [087A]
440 GOTO 400                              [C74A]
450 PEN 1:PRINT invoff$;:LOCATE a(wahl),
    b(wahl):PRINT menue$(wahl);invon$;:R
    ETURN                                 [4D6C]
460 :                                     [CCE8]
470 IF wahl=3 AND flag=0 THEN GOTO 620    [A900]
480 IF wahl=6 AND flag=0 THEN CLS:GOSUB
    2390:IF dis=5 THEN 490 ELSE FOR t=1
    TO 2000:NEXT t:CLS:GOTO 350           [C31E]
490 IF wahl=6 AND flag=0 THEN CLS:WINDOW
     SWAP 1,0:LOCATE 18,24:PRINT "Direct
    ory ";:WINDOW SWAP 1,0:PEN 3:CLS:CAT
    :GOSUB 1910:CLS:GOTO 350              [A4F8]
500 IF wahl=1 AND flag=0 THEN GOTO 760    [0DFA]
510 IF wahl=2 AND flag=0 THEN GOTO 1160   [0654]
520 IF wahl=2 AND flag=2 THEN CLS:GOTO 3
    50                                    [9232]
530 IF wahl=1 AND flag=2 THEN 1220        [9DA2]
540 IF flag=3 THEN GOTO 750               [8FAC]
550 IF wahl=4 AND flag=0 THEN GOTO 1320   [365C]
560 IF flag=4 AND wahl=2 THEN CLS:GOTO 3
    50                                    [5A3E]
570 IF flag=4 AND wahl=1 THEN 1380        [9ABC]
580 IF wahl=5 AND flag=0 THEN CLS:GOTO 1
    540                                   [83A4]
590 IF wahl=2 AND flag=5 THEN CLS:GOTO 3
    50                                    [3E46]
600 IF wahl=1 AND flag=5 THEN CLS:GOTO 1
    600                                   [BC92]
610 GOTO 1950                             [3FBE]
620 '-------------                        [F2F6]
630 '- Zeitraum -                         [8696]
640 '-------------                        [DCFA]
650 CLS:WINDOW SWAP 1,0:LOCATE 18,24:PRI
    NT " Zeitraum ";:WINDOW SWAP 1,0:RES
    TORE 1860:FOR t=1 TO 12:READ menue$(
    t):a$=menue$(t):READ a(t),b(t):x=a(t
    ):y=b(t):GOSUB 1780:NEXT t            [C3AA]
660 x=16:y=16:a$=" *Menue* ":GOSUB 1780   [E34E]
670 MOVE 222,128:DRAWR 0,190,3            [1840]
680 MOVE 398,128:DRAWR 0,190,3            [D75E]
690 FOR t=1 TO 3:MOVE 48,128+(48*t):DRAW
    R 526,0,2:MOVE 48,126+(48*t):DRAWR (
    33*16)-2,0,2:NEXT t                   [7192]
700 MOVE 224,128:DRAWR 172,0,2:MOVE 224,
    126:DRAWR 172,0,2                     [346E]
710 LOCATE 4,2:PEN 2:PRINT CHR$(214);STR
    ING$(32,143);CHR$(212);invon$;CHR$(8
    );:PEN 1:PRINT CHR$(214);invoff$      [CAC2]
720 FOR t=3 TO 13:LOCATE 37,t:PEN 1:PRIN
    T CHR$(143);:NEXT t:LOCATE 37,14:PRI
    NT CHR$(212);                         [7A48]
730 LOCATE 26,15:PEN 1:PRINT CHR$(143);:
    LOCATE 26,16:PRINT CHR$(143);:LOCATE
     26,17:PRINT CHR$(212);               [F56C]
740 flag=3:px=3:py=4:wahl=1:flag=3:GOTO
    400                                   [2C54]
750 mona$=menue$(wahl):ma=wahl:GOSUB 219
    0:GOTO 1950                           [D7F8]
760 '-------------                        [0100]
770 '- Eingeben -                         [3B38]
780 '-------------                        [2704]
790 eing=1                                [EEA2]
800 WINDOW SWAP 0,1:LOCATE 18,24:PRINT i
    nvoff$;" Eingeben ";:WINDOW SWAP 0,1
    :CLS                                  [CD0A]
810 PRINT CHR$(150);STRING$(38,154);CHR$
    (156);:LOCATE 8,1:PRINT CHR$(158);    [E946]
820 FOR t=2 TO 18:LOCATE 1,t:PRINT CHR$(
    149);TAB(8);CHR$(149);:LOCATE 40,t:P
    RINT CHR$(149);:NEXT t:LOCATE 8,19:P
    RINT CHR$(155)                        [8328]
830 LOCATE 1,19:PRINT CHR$(147);STRING$(
    38,154);CHR$(153);                    [2C90]
840 blatt=0:GOSUB 860:'Erstes Blatt       [594C]
850 GOTO 950                              [B468]
860 IF blatt=1 THEN kont=(mona(ma)+17):k
    ont=(kont MOD 7) ELSE kont=mona(ma)+
    1                                     [EE4E]
870 IF kont MOD 7>7 THEN kont=((kont MOD
     7) MOD 7)                            [62D4]
880 FOR t=1 TO 16:LOCATE 2,t+1:IF kont=8
     THEN kont=1                          [43CC]
890 IF blatt*16+t>monla(ma) THEN PRINT "
    <6>":GOTO 920                         [BCF2]
900 IF kont=0 THEN kont=7                 [2160]
910 PRINT LEFT$(day$(kont),2);" ";USING
    "##";blatt*16+t;:PRINT ".";:kont=kon
    t+1:LOCATE 9,t+1:PRINT LEFT$(eintr$(
    ma,(blatt*16)+t),32);                 [0E7A]
920 NEXT t                                [381C]
930 IF blatt=1 THEN LOCATE 2,17:PRINT "<
    6>";:LOCATE 9,17:PRINT "<30>";        [6350]
940 RETURN                                [A23A]
950 'eing=1                               [6AEC]
```

Listing. Alle Termine fest im Griff

```
960 t$=eintr$(ma,(blatt*16+eing))+STRING
    $(31-LEN(eintr$(ma,(blatt*16)+eing))
    ,128)                                  [2A86]
970 LOCATE 9,eing+1:PRINT invoff$;i$;t$;
    i$                                     [87E4]
980 IF INKEY(0)=0 THEN 1080                [5F2C]
990 y$=INKEY$                              [2380]
1000 IF INKEY(2)=0 THEN 1100               [9F62]
1010 IF INKEY(3)=128 THEN IF INKEY(23)=1
     28 THEN CLS:GOTO 350                  [962A]
1020 IF INKEY(11)=128 THEN IF INKEY(23)=
     128 THEN eintr$(ma,(blatt*16+eing))
     ="":GOTO 960                          [B886]
1030 IF INKEY(4)=128 THEN IF INKEY(23)=1
     28 AND blatt=0 THEN blatt=1:eing=1:
     CLS#2:GOSUB 860:GOSUB 1070:GOTO 950
                                           [C45C]
1040 IF INKEY(12)=128 THEN IF INKEY(23)=
     128 AND blatt=1 THEN blatt=0:eing=1
     :CLS#2:GOSUB 860:GOSUB 1070:GOTO 95
     0                                     [82BC]
1050 IF INKEY(9)=0 THEN 1120               [AF7E]
1060 GOTO 960                              [85BE]
1070 LOCATE 9,eing+1:PRINT t$;:RETURN      [DCF0]
1080 IF eing=1 THEN GOTO 960               [7D20]
1090 GOSUB 1070:eing=eing-1:GOTO 960       [1A3E]
1100 IF blatt=0 AND eing=16 OR blatt=1 A
     ND eing=monla(ma)-16 THEN 960         [3940]
1110 GOSUB 1070:eing=eing+1:GOTO 960       [042C]
1120 LOCATE 9,eing+1:PRINT "<31>";         [7DCC]
1130 IF LEN(eintr$(ma,eing))>30 THEN PRI
     NT CHR$(7);:LOCATE 9,eing:PRINT "Di
     eses Feld ist voll !!";:FOR t=1 TO
     2000:NEXT t:GOSUB 1070:GOTO 950       [FD90]
1140 GOSUB 20000:eingabe$=b$:IF LEN (ein
     gabe$)>30 THEN 1130                   [D6C2]
1150 eintr$(ma,eing+blatt*16)=eingabe$:W
     INDOW#0,1,40,4,22:GOTO 950            [761A]
1160 '-----------                          [35FC]
1170 '- Drucken -                          [55EC]
1180 '-----------                          [3A00]
1190 CLS:WINDOW SWAP 1,0:LOCATE 18,24:PR
     INT " Drucken<2>";:WINDOW SWAP 1,0    [2240]
1200 DATA " Drucken ",11,10," *Menue* ",
     22,10                                 [C546]
1210 flag=2:RESTORE 1200:FOR t=1 TO 2:RE
     AD menue$(t),a(t),b(t):a$=menue$(t)
     :x=a(t):y=b(t):GOSUB 1780:MOVE 318,
     176:DRAWR 0,46,3:NEXT:px=2:py=1:wah
     l=1:GOTO 400                          [48CA]
1220 IF INP(&F500)=90 THEN LOCATE 1,17:P
     RINT "<2>Bitte schalten sie den Dru
     cker ein !";:PRINT CHR$(7);:GOTO 12
     20                                    [ADAE]
1230 CLS                                   [1690]
1240 PRINT#8,CHR$(27);"x";CHR$(1);         [144C]
1250 PRINT#8,STRING$(40,"*"):PRINT#8,"*
     Terminausdruck fuer";TAB(23);mona$;
     TAB(32);jahr;TAB(40);"*":PRINT#8,ST
     RING$(40,"*");:PRINT#8:PRINT#8        [0B8C]
1260 druck=mona(ma)+1:IF druck=8 THEN dr
     uck=1                                 [5EA4]
1270 FOR t=1 TO monla(ma)                  [2DD8]
1280 PRINT#8,day$(druck);TAB(12);:PRINT#
     8,USING "##";t;:PRINT#8,".";TAB(18)
     ;eintr$(ma,t)                         [0100]
1290 druck=druck+1:IF druck=8 THEN druck
     =1                                    [9348]
1300 NEXT t                                [A26E]
1310 PRINT#8:PRINT#8,STRING$(40,"*"):PRI
     NT#8,"* Terminkalender * By Patrick
      Schuster *":PRINT#8,STRING$(40,"*"
     ):GOTO 1160                           [0C6E]
1320 '---------                            [E344]
1330 '- Laden -                            [D118]
1340 '---------                            [E348]
1350 CLS:WINDOW SWAP 1,0:LOCATE 18,24:PR
     INT "<2>Laden<3>";:WINDOW SWAP 1,0    [10EC]
1360 DATA "<2>Laden<2>",11,10," *Menue*
     ",22,10                               [6704]
1370 flag=4:RESTORE 1360:FOR t=1 TO 2:RE
     AD menue$(t),a(t),b(t):a$=menue$(t)
     :x=a(t):y=b(t):GOSUB 1780:MOVE 318,
     176:DRAWR 0,46,3:NEXT:px=2:py=1:wah
     l=1:GOTO 400                          [B8EA]
1380 LOCATE 10,13:PRINT invoff$;"Welches
      File -->&lt;4>";:LOCATE 27,13:LINE IN
     PUT "",wahl$                          [6B44]
1390 PRINT invoff$                         [D98C]
1400 IF LEN(wahl$)>8 THEN 1380             [E58A]
1410 GOSUB 2390:IF dis=5 THEN 1430         [1A4C]
1420 FOR t=1 TO 2000:NEXT t:CLS:GOTO 132
     0                                     [5AEE]
1430 OPENIN wahl$                          [DA42]
1440 INPUT#9,jahr,ma                       [8940]
1450 FOR t=1 TO 7:INPUT#9,day$(t),day(t)
     :NEXT t                               [CD44]
1460 INPUT#9,mona$                         [F4A4]
1470 FOR t=1 TO 12                         [B036]
1480 FOR r=1 TO 32                         [8F38]
1490 INPUT#9,eintr$(t,r)                   [035E]
1495 IF EOF THEN CLOSEIN:GOTO 1530         [66A8]
1500   NEXT r                              [CBEE]
1505 IF EOF THEN CLOSEIN:GOTO 1530         [1798]
1510 NEXT t                                [CA74]
1520 CLOSEIN                               [79EA]
1530 GOSUB 130:WINDOW SWAP 1,0:LOCATE 33
     ,24:PRINT invoff$;jahr;:WINDOW SWAP
      1,0:LOCATE 10,12:PRINT "<41>";:GOS
     UB 2190:GOTO 1360                     [5928]
1540 '-------------                        [76B4]
1550 '- Speichern -                        [379A]
1560 '-------------                        [AEB8]
1570 CLS:WINDOW SWAP 1,0:LOCATE 18,24:PR
     INT "Speichern ";:WINDOW SWAP 1,0     [F56E]
1580 DATA Speichern,11,10," *Menue* ",22
     ,10                                   [8EFE]
1590 flag=5:RESTORE 1580:FOR t=1 TO 2:RE
     AD menue$(t),a(t),b(t):a$=menue$(t)
     :x=a(t):y=b(t):GOSUB 1780:MOVE 318,
     176:DRAWR 0,46,3:NEXT:px=2:py=1:wah
     l=1:GOTO 400                          [D8FC]
1600 LOCATE 1,13:PRINT invoff$;"Name der
      abgespeicherten Datei :";jahr        [8AF0]
1610 GOSUB 2390:IF dis=5 THEN 1630         [4A54]
1620 FOR t=1 TO 2000:NEXT t:CLS:GOTO 154
     0                                     [BAFA]
1630 wahl$=STR$(jahr):OPENOUT wahl$        [87BC]
1640 PRINT#9,jahr,ma                       [7C3E]
1650 FOR t=1 TO 7:PRINT#9,day$(t),day(t)
     :NEXT t                               [9842]
1660 PRINT#9,mona$                         [81A2]
1670 FOR t=1 TO 12                         [403A]
1680 FOR r=0 TO 31                         [8D38]
1690 PRINT #9,eintr$(t,r)                  [DB9C]
1700   NEXT r                              [B1F2]
1710 NEXT t                                [B478]
1720 CLOSEOUT                              [52B0]
1730 LOCATE 1,13:PRINT "<39>":GOSUB 2210
     :GOTO 1580                            [501E]
1740 :                                     [CD4C]
1750 '***********************************
     *                                     [2EA4]
1760 '* Sub. Print Menue Icons (x,y,a$)
     *                                     [2F46]
1770 '***********************************
     *                                     [D6A8]
1780 LOCATE x-1,y-1:PRINT invon$;:PEN 2:
     PRINT CHR$(215);CHR$(8);:PEN 3:PRIN
     T CHR$(213);STRING$(LEN(a$),143);CH
     R$(212);CHR$(8);:PEN 2:PRINT CHR$(2
     14);                                  [D358]
1790 LOCATE x-1,y:PEN 2:PRINT CHR$(143);
     :PEN 1:PRINT A$;:PEN 2:PRINT CHR$(1
     43);                                  [465E]
1800 LOCATE x-1,y+1:PEN 2:PRINT CHR$(212
     );CHR$(8);:PEN 3:PRINT CHR$(214);ST
     RING$(LEN(a$),143);CHR$(215);CHR$(8
     );:PEN 2:PRINT CHR$(213);             [4566]
1810 RETURN                                [BF94]
1820 '***********************************
     *                                     [C9A0]
1830 '* Sub. Data  Menue Icons (x,y,a$)
     *                                     [B05C]
1840 '***********************************
     *                                     [B9A4]
1850 DATA " Eingeben ",6,4,Ausdrucken,6,
     10," Zeitraum ",6,16,"<2>Laden<2>",
     25,4,Speichern,25,10,Directory,25,1
     6                                     [8AD8]
1860 DATA " Januar<2>",5,4," Februar ",5
     ,7,"<2>Maerz<2>",5,10,"<2>April<2>"
     ,5,13                                 [EAC0]
1870 DATA "<3>Mai<3>",16,4,"<2>Juni<3>",
     16,7,"<2>Juli<3>",16,10," August<2>
     ",16,13                               [3808]
1880 DATA "September",27,4," Oktober ",2
     7,7,"November ",27,10,"Dezember ",2
     7,13                                  [4CB8]
1890 DATA Montag,Dienstag,Mittwoch,Donne
     rstag,Freitag,Samstag,Sonntag         [D532]
1900 '***** Warteschleife                  [B692]
1910 LOCATE 4,18:PEN 1:PRINT "** Bitte d
     ruecken sie die COPY-Taste **";       [B5DA]
1920 IF INKEY(9)=0 THEN LOCATE 4,18:PRIN
     T "<37>";:RETURN                      [7F84]
1930 GOTO 1920                             [9F24]
1940 '===================================
     ========                              [0B2E]
1950 '= Sub. Inkeyroutiene fuer Sub. Ico
     n I/II =                              [F0A0]
1960 '= Bewegung mit Cursor / Auswahl mi
     t COPY =                              [3A2C]
1970 '===================================
     ========                              [3834]
1980 in=11                                 [80D0]
1990 IF rag=1 AND INKEY(0)=0 THEN rag=0:
     PRINT invoff$;:LOCATE 16,16:PRINT m
     $;::in=11:LOCATE 16,13:PRINT CHR$(2
     4);" August<2>";CHR$(24);invon$:GOT
     O 2060                                [9F5E]
```

```
2000 IF rag=1 THEN PRINT invoff$;:LOCATE
      16,13:PRINT " August<2>";CHR$(24);
     :LOCATE 16,16:PRINT m$;CHR$(24);inv
     on$:in=11:GOTO 2060                   [9F04]
2010 y$=INKEY$                             [CEC2]
2020 IF INKEY(0)=0 THEN in=0:GOTO 2100     [CA14]
2030 IF INKEY(2)=0 THEN in=2:GOTO 2120     [5D22]
2040 IF INKEY(8)=0 THEN in=8:GOTO 2150     [B142]
2050 IF INKEY(1)=0 THEN in=1:GOTO 2170     [792C]
2060 IF INKEY(9)=0 AND rag=1 THEN CLS:ra
     g=0:GOTO 350                          [F54C]
2070 IF rag<>1 THEN IF INKEY(9)=0 THEN i
     n=9                                   [CC9C]
2080 IF in=11 THEN 1970                    [489E]
2090 GOTO 430                              [A2B6]
2100 IF wahl=1 OR wahl=py+1 OR wahl=py*2
     +1 OR wahl=py*3+1 OR wahl=py*4+1 TH
     EN 400                                [4E04]
2110 GOSUB 450:wahl=wahl-1:GOTO 400        [C8E2]
2120 IF flag=3 AND wahl=8 THEN rag=1:GOS
     UB 450:GOTO 400                       [6906]
2130 IF wahl=py OR wahl=py*2 OR wahl=py*
     3 OR wahl=py*4 OR wahl=py*5 THEN 40
     0                                     [0158]
2140 GOSUB 450:wahl=wahl+1:GOTO 400        [0CE4]
2150 IF wahl<=py THEN 400                  [2850]
2160 GOSUB 450:wahl=wahl-py:GOTO 400       [D55C]
2170 IF wahl>=((px*py)-py)+1 THEN 400      [6CA4]
2180 GOSUB 450:wahl=wahl+py:GOTO 400       [015C]
2190 '                                     [2526]
2200 WINDOW SWAP 1,0:LOCATE 3,24:PRINT m
     ona$;:WINDOW SWAP 1,0:RETURN          [AB80]
2210 '***********************************
     ****************************          [4B1C]
2220 '* Berechnung des 1. jedes Monats u
     nd des Anfangswochentages *           [65E6]
2230 '***********************************
     ****************************          [CB20]
2240 nt=429+INT(365.25*(jahr-1))           [6E7C]
2250 IF nt<694098 THEN nt=nt+1             [258C]
2260 IF nt<621050 THEN nt=nt+1             [6362]
2270 IF nt<584526 THEN nt=nt+1             [3984]
2280 IF nt>=767148 THEN nt=nt-1            [B90E]
2290 IF nt>=803672 THEN nt=nt-1            [9402]
2300 IF nt>=840196 THEN nt=nt-1            [4BF6]
2310 w=(nt-2)/7:w=ROUND(7*(w-INT(w)),0)    [CA7C]
2320 IF jahr MOD 400=0 THEN schaltjahr=-
     1:GOTO 2360                           [52D0]
2330 IF jahr MOD 100=0 THEN schaltjahr=0
     :GOTO 2360                            [1670]
2340 schaltjahr=jahr MOD 4=0               [8960]
2350 DATA 31,28,31,30,31,30,31,31,30,31,
     30,31                                 [3E74]
2360 RESTORE 2350:FOR t=1 TO 12:READ mon
     la(t):NEXT t:monla(2)=28-schaltjahr
                                           [0A1A]
2370 mona(1)=w-1:FOR t=2 TO 12:m=(mona(t
     -1)-1)+monla(t-1):mona(t)=(m MOD 7)
     +1:NEXT t                             [6C80]
2380 RETURN                                [8E9A]
2390 '-------------------                  [D6D8]
2400 '- Laufwerkabfrage -                  [B960]
2410 '-------------------                  [6FCA]
2420 dis=1                                 [652C]
2430 OUT(&FA7E),1:FOR i=1 TO 1000:NEXT:O
     UT(&FB7F),4:OUT(&FB7F),(-PEEK(&A700
     )+2):st=INP(&FB7F):OUT(&FA7E),0:rd=
     st AND 32                             [4132]
2440 IF rd=0 THEN LOCATE 1,19:PRINT "Dis
     kette befindet sich nicht im Laufwe
     rk";:RETURN                           [4DB8]
2450 wp=st AND 64:IF wp=64 THEN LOCATE 2
     ,19:PRINT "<3>Diskette ist Schreibg
     eschuetzt<5>";:RETURN                 [0F98]
2460 dis=5:RETURN                          [EA70]
2470 FOR t=1 TO 1000:NEXT t:CLS:RESUME 3
     50                                    [ADCC]
20000 y=eing+1:x=9                         [63DE]
20010 s=0:b$=""                            [1668]
20020 LOCATE x,y:PRINT CHR$(246);          [9432]
20030 IF x=9 THEN 20050                    [E030]
20040 LOCATE x-1,y:PRINT MID$(b$,s,1);     [9EB8]
20050 a$=INKEY$:IF a$="" THEN GOTO 20050
                                           [3756]
20060 IF INKEY(79)=0 AND x>5 THEN 20130    [AEB0]
20070 IF INKEY(18)=0 THEN 20150            [01AC]
20080 IF ASC(A$)<32 OR ASC(A$)>126 THEN
      20020                                [DD3E]
20090 IF X=39 THEN 20100 ELSE x=x+1:s=s+
      1:LOCATE x,y:PRINT a$;:b$=b$+a$      [D5C2]
20100 GOTO 20020                           [B3C0]
20110 :                                    [179C]
20120 :                                    [069E]
20130 IF x=9 THEN 20140 ELSE x=x-1:s=s-1
      :LOCATE x+1,y:PRINT " ";:LOCATE 9,
      y:b$=LEFT$(b$,s):PRINT b$;" ";       [8EA4]
20140 GOTO 20020                           [2BC8]
20150 RETURN                               [DFF0]
```

Listing. Alle Termine fest im Griff (Schluß)

# Geld regiert die Welt

**Es gibt wohl heutzutage und hierzulande niemanden mehr, der nicht über ein Girokonto verfügt. Nutzen Sie Ihren CPC zur effizienten Kontrolle der ständigen Geldbewegungen.**

Mit einem Girokonto lebt es sich doch sehr angenehm, denn man braucht sich eigentlich um nichts zu kümmern. Daueraufträge, Einzugsermächtigungen und viele Errungenschaften mehr nehmen Ihnen alle Arbeit ab. Aber manchen läßt doch nie das Gefühl los, er müsse sich von Zeit zu Zeit von der Korrektheit seiner Kontoführung selbst überzeugen. Diesem Zweifler geben wir mit »Giro« ein Hilfsmittel an die Hand, das es ihm erlaubt, außer regelmäßigen Kontrollen auch Vorausplanungen leicht und schnell durch seinen CPC ausführen zu lassen. Da das Programm komplett über Menüs dialoggesteuert abläuft, beschränken wir hier die Ausführungen auf ein notwendiges Minimum. Beim ersten Programmstart ist zunächst die Eingabe von Fixbuchungen notwendig. Alle anderen Funktionen bleiben vorher gesperrt. Zusätzlich fordert Giro später von Ihnen die Eingabe der laufenden Einnahmen beziehungsweise Ausgaben. Jeweils am Monatsanfang geben Sie dem Programm Ihr Gehalt als Eingang bekannt, worauf es dann die Fixbuchungen listet und gegenrechnet. Bei Anzeige des

```
Kontofuehrung

1. Eingabe Gehalt
2. Eingabe Abzuege/Bezuege
3. Buchungssaetze fest
4. Anzeige Kontostand
5. Konto-Statistik
6. Vorschau
7. Datensicherung
8. Ende der Arbeit

Auswahl
```

Die Menüpunkte von »Giro«

```
Vorschau ueber 3 Monate

Oktober                      November                     Dezember
Miete          1400.00 DM    Miete          1400.00 DM    Miete          1400.00 DM
Lebensversich    92.00 DM    Lebensversich    92.00 DM    Lebensversich    92.00 DM
Hausratversic   189.00 DM    Heizkosten      265.00 DM    Alimente        570.00 DM
Alimente        570.00 DM    Alimente        570.00 DM

Belastung:     2251.00 DM    Belastung:     2327.00 DM    Belastung:     2062.00 DM

In dieser Vorschau werden nur die fest angelegten Buchungssaetze verwendet !!
```

aktuellen Kontostandes erscheint im rechten Teil des Bildschirms zusätzlich der Vormonat zum Vergleich. Insgesamt passen 17 monatliche Buchungen auf den Bildausschnitt. Selbstverständlich können Sie auch mit mehr Buchungen arbeiten; dann ist jedoch erforderlich, per Tastendruck auf die nächste Bildschirm-Seite zu blättern. Für den schnellen Überblick sorgt eine Statistikfunktion mit Anzeige der bislang aufgelaufenen Beträge einzelner Fixbuchungen. Besonders interessant ist sicherlich die Vorschau auf die nächsten drei Monate. Bei dieser Berechnung sind natürlich nur die Fixbuchungen berücksichtigt. Damit Sie nicht bei jeder Benutzung des Programms alles neu eingeben müssen, läßt sich der erfaßte Datenbestand auf Datenträger sichern – das ist dringend anzuraten, bevor Sie mit dem Menüpunkt 8 (Ende der Arbeit) den Programmlauf beenden. (Ingo Strecker/ja)

| Steckbrief | |
|---|---|
| Programm: | Giro |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

```
10 '*****************************         [E828]
20 '*      Copyright by            *       [9D62]
30 '*                              *       [77FC]
40 '*      Ingo Strecker           *       [A79E]
50 '*      Haigerlocherstr.55      *       [66FC]
60 '*      7450 Hechingen          *       [B674]
70 '*                              *       [9404]
80 '*      Tel. 07471/13197        *       [B55A]
90 '*****************************         [103B]
100 GOTO 400                              [E43C]
102 '****************************         [7538]
104 '   Kontostandsanzeige                [906E]
106 '****************************         [7140]
110 DATA Januar,Februar,Maerz,April,Mai,
    Juni,Juli,August,September,Oktober,N
    ovember,Dezember                      [4024]
120 l1=4:MODE 2:PRINT CHR$(24);:PRINT SP
    ACE$(80):LOCATE 1,1:PRINT"<2>Kontost
    and von "a1$" am "b$"":PRINT CHR$(24
    )                                     [0020]
130 IF b$="" THEN GOTO 440                [D894]
140 WINDOW#1,2,38,3,21:WINDOW#2,43,79,3,
    21:PLOT 0,373:DRAWR 312,0:DRAWR 0,-3
    12:DRAWR -312,0:DRAWR 0,312:PLOT 327
    ,373:DRAWR 312,0:DRAWR 0,-312:DRAWR
    -312,0:DRAWR 0,312                    [FAD8]
150 RESTORE:FOR k=1 TO VAL(MID$(b$,4,2))
    :READ f$:NEXT                         [052E]
160 RESTORE:FOR k=1 TO VAL(MID$(b1$,4,2)
    ):READ f1$:NEXT                       [7CF4]
170 IF b1$="" THEN 280                    [5050]
180 LOCATE#2,15,1:PRINT#2,CHR$(24)" "f1$
    " "CHR$(24):PRINT#2,"Kontostand:":LO
    CATE#2,27,2:PRINT#2,USING"#####.## D
    M";ialt                               [6EC0]
190 LOCATE#2,1,11:PRINT#2,"Gehalt":LOCAT
    E#2,27,11:PRINT#2,USING"#####.## DM"
    ;balt:FOR k=1 TO falt:IF e1(k)=0 THE
    N l1=l1+1:LOCATE#2,1,l1:PRINT#2,calt
    $(k):LOCATE#2,27,l1:PRINT#2,USING"##
    ###.## DM";calt(k)                    [7B48]
200 IF e1(k)=VAL(MID$(b1$,4,2)) THEN l1=
    l1+1:LOCATE#2,1,l1:PRINT#2,calt$(k):
    LOCATE#2,27,l1:PRINT#2,USING"#####.#
    # DM";calt(k)                         [CA12]
210 NEXT                                  [6CE4]
220 IF halt+l1>17 AND a=4 THEN LOCATE#2,
    1,l1+2:PRINT#2,CHR$(24)" zum Blaette
    rn Taste "CHR$(24):CALL &BB06 ELSE 2
    40                                    [30BA]
230 CLS#2:l1=3:LOCATE#2,15,1:PRINT#2,CHR
    $(24)" "f1$" "CHR$(24):PRINT#2,"Kont
    ostand:":LOCATE#2,27,2:PRINT#2,USING
    "#####.## DM";ialt                    [AB28]
240 FOR k=1 TO halt:l1=l1+1:LOCATE#2,1,l
    1:PRINT#2,ealt$(k):IF LEFT$(galt$(k)
    ,1)="-" THEN LOCATE#2,26,l1:PRINT#2,
    "-":LOCATE#2,27,l1:PRINT#2,USING"###
    ##.## DM";VAL(MID$(galt$(k),2)):GOTO
     260 ELSE LOCATE#2,26,l1:PRINT#2,"+"
                                          [59F8]
250 LOCATE#2,27,l1:PRINT#2,USING"#####.#
    # DM";VAL(galt$(k))                   [D5FE]
260 NEXT                                  [64EE]
270 LOCATE#2,1,19:PRINT#2,"Kontostand ne
    u:":LOCATE#2,27,19:PRINT#2,USING"###
    ##.## DM";i;                          [5386]
280 l=4:LOCATE#1,15,1:PRINT#1,CHR$(24)"
    "f$" "CHR$(24):PRINT#1,"Kontostand:"
    :LOCATE#1,27,2:PRINT#1,USING"#####.#
    # DM";i:IF a=1 THEN RETURN            [0010]
290 LOCATE#1,1,l:PRINT#1,"Gehalt":LOCATE
    #1,27,l:PRINT#1,USING"#####.## DM";b
    :FOR k=1 TO f:IF e1(k)=0 THEN l=l+1:
    LOCATE#1,1,l:PRINT#1,c$(k):LOCATE#1,
    27,l:PRINT#1,USING"#####.## DM";c(k)
                                          [9BE6]
300 IF e1(k)=VAL(MID$(b$,4,2)) THEN l=l+
    1:LOCATE#1,1,l:PRINT#1,c$(k):LOCATE#
    1,27,l:PRINT#1,USING"#####.## DM";c(
    k)                                    [021E]
310 NEXT                                  [63E6]
320 IF h+l>17 THEN LOCATE#1,1,l+2:PRINT#
    1,CHR$(24)" zum Blaettern Taste "CHR
    $(24):CALL &BB06:CLS#1:l=3:flag1=1:L
    OCATE#1,15,1:PRINT#1,CHR$(24)" "f$"
    "CHR$(24):PRINT#1,"Kontostand:":LOCA
    TE#1,27,2:PRINT#1,USING"#####.## DM"
    ;i                                    [6870]
330 FOR k=1 TO h:l=l+1:LOCATE#1,1,l:PRIN
    T#1,e$(k):IF LEFT$(g$(k),1)="-" THEN
     LOCATE#1,26,l:PRINT#1,"-":LOCATE#1,
    27,l:PRINT#1,USING"#####.## DM";VAL(
    MID$(g$(k),2)):GOTO 350 ELSE LOCATE#
    1,26,l:PRINT#1,"+"                    [E194]
340 LOCATE#1,27,l:PRINT#1,USING"#####.##
     DM";VAL(g$(k))                       [3616]
350 NEXT                                  [6BEE]
360 LOCATE#1,1,19:PRINT#1,"Kontostand ne
    u:":LOCATE#1,27,19:PRINT#1,USING"###
    ##.## DM";j;:IF flag1<>1 THEN 390     [F854]
370 LOCATE#1,1,l+2:PRINT#1,CHR$(24)" Bla
    ettern ?? (J/N) "CHR$(24):flag1=0     [6278]
380 d$=INKEY$:IF d$="" THEN 380 ELSE IF
    LOWER$(d$)="j" THEN CLS#1:GOTO 280    [1FF6]
390 RETURN                                [AA38]
392 '****************************         [74A2]
394 '   Initialisierung                   [E6EE]
396  ****************************         [7CAA]
400 f=0:h=0:INK 2,24:DIM c$(20),c(20),cs
    (20),d(20),e(20),e$(20),g$(20),e1(20
    ),ealt$(20),galt$(20),b2$(20),b2alt$
    (20),calt$(20),calt(20)               [996C]
410 MODE 1:PRINT CHR$(24)"<3>Anmeldung z
    um Programm<2>K O N T O<4>"CHR$(24):
    LOCATE 1,8:PRINT"Konto-Inhaber:":LOC
    ATE 1,13:PRINT"Datum (TTMMJJ):"       [0348]
415 LOCATE 1,23:PRINT CHR$(24)" Bei Neua
    nlage eines Kontos:"SPACE$(12)CHR$(2
    4);:PRINT:PRINT"Konto-Inhaber: neuan
    lage"                                 [7C4E]
420 LOCATE 17,8:INPUT"",a$:LOCATE 17,13:
    INPUT"",b3$:b3$=MID$(b3$,1,2)+"."+MI
    D$(b3$,3,2)+"."+MID$(b3$,5,2)         [1F4A]
430 IF LOWER$(a$)<>"neuanlage" THEN GOSU
    B 1290 ELSE CLS:PRINT CHR$(24)"<6>Ne
    uanlage einer Konto-Datei<7>"CHR$(24
    ):LOCATE 1,8:PRINT"Konto-Inhaber:":L
    OCATE 1,13:PRINT"Kontostand alt:":LO
    CATE 17,8:INPUT"",a$:LOCATE 17,13:IN
    PUT"",j:a1$=a$:flag=1                 [62E8]
432 '****************************         [B498]
434 '   Hauptmenue                        [98BC]
436 '****************************         [ACA0]
440 MODE 1:PRINT STRING$(40,"-")CHR$(24)
    TAB(13)"Kontofuehrung"SPACE$(15)CHR$
    (24)STRING$(40,"-")                   [CF4A]
450 LOCATE 1,6:PRINT"1. Eingabe Gehalt":
    PRINT:PRINT"2. Eingabe Abzuege/Bezue
    ge":PRINT:PEN 2:PRINT"3. Buchungssae
    tze fest":PEN 1:PRINT:PRINT"4. Anzei
    ge Kontostand":PRINT                  [7928]
460 PRINT"5. Konto-Statistik":PRINT:PRIN
    T"6. Vorschau":PRINT:PRINT"7. Datens
    icherung":PRINT                       [BCF2]
470 PRINT"8. Ende der Arbeit"             [0DC0]
480 IF flag=1 THEN INK 2,0,24             [6708]
490 LOCATE 1,23:PRINT"Auswahl":LOCATE 1,
    25:INPUT"",a:IF flag=1 AND a=3 THEN
    INK 2,24 ELSE IF flag=0 THEN 500 ELS
    E 480                                 [CBB4]
500 IF a<1 OR a>8 THEN 480                [8BEA]
510 ON a GOSUB 530,640,810,1040,1490,162
    0,1120,1480                           [E400]
520 GOTO 440                              [F850]
522 '****************************         [5598]
524 '   Gehaltseingebe                    [63B2]
526 '****************************         [8DA0]
```

```
530 b1$=b$:b$=b3$:halt=h:ialt=i:i=j:balt
    =b:falt=f                                [0288]
540 FOR f1alt=1 TO falt:calt$(f1alt)=c$(
    f1alt):calt(f1alt)=c(f1alt):NEXT        [09E2]
550 FOR h1alt=1 TO halt:ealt$(h1alt)=e$(
    h1alt):galt$(h1alt)=g$(h1alt):b2alt$
    (h1alt)=b2$(h1alt):NEXT                 [1EA0]
560 h=0:GOSUB 120                            [3DFA]
570 WINDOW#4,1,80,23,25:PRINT#4,CHR$(24)
    :CLS#4:LOCATE#4,2,2:INPUT#4,"Gehalt:
     ",b:PRINT#4,CHR$(24):j=i+b             [7042]
580 LOCATE#1,1,l:PRINT#1,"Gehalt":LOCATE
    #1,27,l:PRINT#1,USING"#####.## DM";b
    :bs=bs+b:FOR k=1 TO f:IF e(k)=0 THEN
     l=l+1:LOCATE#1,1,l:PRINT#1,c$(k):LO
    CATE#1,27,l:PRINT#1,USING"#####.## D
    M";c(k):j=j-c(k):cs(k)=cs(k)+c(k)       [BEDC]
590 IF e(k)=VAL(MID$(b$,4,2)) THEN l=l+1
    :e1(k)=e(k):e(k)=e(k)+d(k):LOCATE#1,
    1,l:PRINT#1,c$(k):LOCATE#1,27,l:PRIN
    T#1,USING"#####.## DM";c(k):j=j-c(k)
    :cs(k)=cs(k)+c(k):IF e(k)>12 THEN e(
    k)=e(k)-12                               [9CF8]
600 NEXT                                     [7CEA]
610 LOCATE#1,1,19:PRINT#1,"Kontostand ne
    u:":LOCATE#1,27,19:PRINT#1,USING"###
    ##.## DM";j;                             [317C]
620 CALL &BB06                               [B308]
630 RETURN                                   [DD32]
632 '******************************          [949C]
634 '   Abzuege/Bezuege eingeben             [40B4]
636 '******************************          [5CA4]
640 MODE 1:PRINT CHR$(24)"<5>*** Auswahl
    moeglichkeiten ***<6>"CHR$(24):PRINT
    :PRINT:PRINT"1. Abzuege eingeben":PR
    INT:PRINT"2. Bezuege eingeben":PRINT
    :PRINT"3. Hauptmenue"                   [B042]
650 LOCATE 1,23:PRINT"Auswahl":LOCATE 1,
    25:INPUT"",a1:IF a1=3 THEN 440 ELSE
    IF a1<1 OR a1>3 THEN 650                 [DB9C]
660 GOSUB 120:WINDOW#3,1,80,23,25            [A4D8]
670 h=h+1:PRINT#3,CHR$(24):CLS#3:PRINT#3
    ,"<2>Buchungstext:":PRINT#3,"<2>Buch
    ungsbetrag:":PRINT#3,"<2>Weiter buch
    en ?? (J/N):":LOCATE#3,44,1:PRINT#3,
    "A N L E G E N":LOCATE#3,44,2:PRINT#
    3,"Buchungstext: a<3>--> Satz aender
    n"                                       [E4FC]
680 LOCATE#3,44,3:PRINT#3,"Buchungsdatum
    : "b3$"":b2$(h)=MID$(b3$,1,2)+MID$(b
    3$,4,2)+MID$(b3$,7,2):LOCATE#3,19,1:
    INPUT#3,"",e$(h)                         [EE8C]
690 IF e$(h)=LOWER$("a") THEN h=h-1:CLS#
    3:PRINT#3,"<2>Buchungstext:<3>"e$(h)
    "":IF LEFT$(g$(h),1)="-" THEN PRINT#
    3,"<2>Buchungsbetrag:"g$(h)"" ELSE P
    RINT#3,"<2>Buchungsbetrag:"+"+"g$(h)
    "":ELSE 760                              [1092]
700 LOCATE#3,44,1:PRINT#3,"Buchungstext:
    ":LOCATE#3,44,2:PRINT#3,"Buchungsbet
    rag:":LOCATE#3,60,1:INPUT#3,"",e1$:L
    OCATE#3,60,2:INPUT#3,"",g1$:IF e1$<>
    "" THEN e$(h)=e1$                        [2138]
710 IF g1$<>"" AND LEFT$(g$(h),1)="-" TH
    EN j=j+VAL(MID$(g$(h),2))-VAL(g1$):g
    $(h)="-"+g1$ ELSE IF g1$<>"" AND LEF
    T$(g$(h),1)<>"-" THEN j=j-VAL(g$(h))
    +VAL(g1$):g$(h)=g1$                      [8ED6]
720 LOCATE#1,1,l:PRINT#1,SPACE$(39):LOCA
    TE#1,1,l:PRINT#1,e$(h):LOCATE#1,27,l
    :PRINT#1,USING"#####.## DM";VAL(MID$
    (g$(h),2)):IF LEFT$(g$(h),1)="-" THE
    N LOCATE#1,26,l:PRINT#1,"-":GOTO 740
     ELSE LOCATE#1,26,l:PRINT#1,"+"         [730C]
730 LOCATE#1,27,l:PRINT#1,USING"#####.##
     DM";VAL(g$(h))                          [6516]
740 LOCATE#1,1,19:PRINT#1,"Kontostand ne
    u:":LOCATE#1,27,19:PRINT#1,"<11>";:L
    OCATE#1,27,19:PRINT#1,USING"#####.##
     DM";j;:PRINT#3,CHR$(24)                 [72B4]
750 CALL &BB06:GOTO 640                      [436A]
760 LOCATE#3,19,2:INPUT#3,"",g$(h):LOCAT
    E#3,26,3:INPUT#3,"",d$:PRINT#3,CHR$(
    24);                                     [F7F4]
770 l=l+1:IF a1=1 THEN j=j-VAL(g$(h)):g$
    (h)="-"+g$(h) ELSE IF a1=2 THEN j=j+
    VAL(g$(h))                               [8828]
780 IF LEFT$(g$(h),1)="-" THEN LOCATE#1,
    26,l:PRINT#1,"-":LOCATE#1,1,l:PRINT#
    1,e$(h):LOCATE#1,27,l:PRINT#1,USING"
    #####.## DM";VAL(MID$(g$(h),2)):GOTO
     800 ELSE LOCATE#1,26,l:PRINT#1,"+"     [9A74]
790 LOCATE#1,1,l:PRINT#1,e$(h):LOCATE#1,
    27,l:PRINT#1,USING"#####.## DM";VAL(
    g$(h))                                   [27AA]
800 LOCATE#1,1,19:PRINT#1,"Kontostand ne
    u:":LOCATE#1,27,19:PRINT#1,USING"###
    ##.## DM";j;:IF LOWER$(d$)="n" THEN
    CALL &BB06:GOTO 640 ELSE 670             [1C16]
802 '******************************          [7B9A]
804 '   Fixbuchungen anlegen                 [AA46]
806 '******************************          [83A2]
810 MODE 1:PRINT CHR$(24)"<5>*** Auswahl
    moeglichkeiten ***<6>"CHR$(24):PRINT
    :PRINT:PRINT"1. Buchungssatz anlegen
    ":PRINT:PRINT"2. Buchungssatz aender
    n":PRINT:PRINT"3. Buchungssatz loesc
    hen":PRINT                               [66E4]
815 PRINT"4. Buchungssatz anzeigen":PRIN
    T:PRINT"5. Hauptmenue"                   [BDE8]
820 LOCATE 1,23:PRINT"Auswahl":LOCATE 1,
    25:INPUT"",a:IF a<1 OR a>5 THEN 820      [0634]
830 ON a GOTO 840,910,980,1010,440           [9766]
840 MODE 1:WINDOW#1,1,40,1,14:WINDOW#2,1
    ,40,15,25:PRINT#1,CHR$(24)" Bitte in
     der Spalte<2>>Zahlungsweise<<<2>di
    e entsprechende Kennzahl eingeben !!
     "CHR$(24):LOCATE#1,1,5:PRINT#1,"1<2
    ><-- monatlich":PRINT#1:PRINT#1,"2<2
    ><-- zweimonatlich":PRINT#1              [DD0E]
850 PRINT#1,"3<2><-- vierteljaehrlich":P
    RINT#1:PRINT#1,"6<2><-- halbjaehrlic
    h":PRINT#1:PRINT#1,"12 <-- jaehrlich
    "                                        [07F0]
860 CLS#2:f=f+1:PRINT#2,STRING$(40,"-"):
    PRINT#2,"Buchungstext:":PRINT#2:PRIN
    T#2,"Buchungsbetrag:":PRINT#2:PRINT#
    2,"Zahlungsweise:":PRINT#2:PRINT#2:P
    RINT#2:PRINT#2,"Anlegen Ende ?? (J/N
    )":LOCATE#2,33,2:PRINT#2,CHR$(24)USI
    NG" Nr.:## ";f:PRINT#2,CHR$(24)          [3C60]
870 LOCATE#2,17,3:INPUT#2,"",c$(f):LOCAT
    E#2,17,5:INPUT#2,"",c(f):LOCATE#2,17
    ,7:INPUT#2,"",d(f)                       [BFBA]
880 IF d(f)<>1 THEN LOCATE#2,1,9:INPUT#2
    ,"naechster Faelligkeitsmonat:",e(f)
    :e1(f)=e(f):LOCATE#2,23,11:INPUT#2,"
    ",d$ ELSE LOCATE#2,23,11:INPUT#2,"",
    d$                                       [E76A]
890 IF flag<>1 THEN j=j-c(f):cs(f)=c(f)      [4534]
900 IF LOWER$(d$)="n" THEN 860:f=f-1 ELS
    E flag=0:GOTO 810                        [F74E]
910 CLS:INPUT"Buchungssatz aendern Nr. "
    ,f1:LOCATE 1,4:PRINT"Buchungstext:<3
    >"c$(f1)"":PRINT:PRINT"Buchungsbetra
    g:"c(f1)"":PRINT:PRINT"Zahlungsweise
    : "d(f1)"":PRINT:PRINT"naechster Fae
    lligkeitsmonat:"e(f1)""                  [CAF2]
920 LOCATE 1,15:PRINT"Buchungstext:":PRI
    NT:PRINT"Buchungsbetrag:":PRINT:PRIN
    T"Zahlungsweise:":PRINT:PRINT"naechs
    ter Faelligkeitsmonat:":LOCATE 17,15
    :INPUT"",c1$:LOCATE 17,17:INPUT"",c1
    :LOCATE 17,19:INPUT"",d1:LOCATE 30,2
    1:INPUT"",e1                             [E37A]
930 IF c1$<>"" THEN c$(f1)=c1$               [0468]
940 IF c1<>0 THEN j=j+c(f1)-c1:cs(f1)=cs
    (f1)-c(f1)+c1:c(f1)=c1                   [5CC2]
950 IF d1<>0 THEN d(f1)=d1                   [DA72]
960 IF e1<>0 THEN e(f1)=e1                   [D07A]
970 GOTO 810                                 [1D64]
980 CLS:INPUT"Buchungssatz loeschen Nr.
    ",f1:LOCATE 1,7:PRINT"Buchungstext:<
    3>"c$(f1)"":PRINT:PRINT"Buchungsbetr
    ag:"c(f1)"":PRINT:PRINT"Zahlungsweis
    e: "d(f1)"":PRINT:PRINT"naechster Fa
    elligkeitsmonat:"e(f1)""                 [08EE]
990 LOCATE 1,19:INPUT"Soll geloescht wer
    den?? (J/N) ",d$:IF LOWER$(d$)="n" T
    HEN 810 ELSE j=j+c(f1):cs(f1)=cs(f1)
    -c(f1)                                   [52AE]
1000 FOR k=f1 TO f:c$(k)=c$(k+1):c(k)=c(
     k+1):d(k)=d(k+1):e(k)=e(k+1):NEXT:f
     =f-1:LOCATE 1,22:PRINT CHR$(24)" S
     a t z<3>i s t<3>g e l o e s c h t "
     CHR$(24):FOR k=1 TO 2000:NEXT:GOTO
     810                                     [AF7C]
1010 MODE 2:PRINT CHR$(24)" Nr.<2>Buchun
     gstext<11>Buchungsbetrag<3>Zahlweis
     e<3>Faell.-Monat "CHR$(24):FOR k=1
     TO f:LOCATE 2,k+2:PRINT USING"##";k
     :LOCATE 7,k+2:PRINT c$(k):LOCATE 31
     ,k+2:PRINT USING"#####.## DM";c(k)     [9018]
1020 LOCATE 50,k+2:PRINT USING"##";d(k):
     LOCATE 64,k+2:PRINT USING"##";e(k):
     NEXT:LOCATE 1,25:PRINT"Weiter mit T
     aste"                                   [D6FC]
1030 d$=INKEY$:IF d$="" THEN 1030 ELSE 8
     10                                      [88E8]
1032 '******************************         [74F2]
1034 '   Kontostand anzeigen                 [EF0A]
1036 '******************************         [7CFA]
1040 GOSUB 120                               [1530]
1050 WINDOW#3,1,80,23,25:PRINT#3,CHR$(24
     ):CLS#3:PRINT#3:PRINT#3,"<2>Buchung
```

**Listing. Immer auf dem laufenden mit »Giro«**

```
     sdatum einblenden ?? (J/N)":PRINT#3
     ,CHR$(24)                              [C7D6]
1060 d$=INKEY$:IF d$="" THEN 1060 ELSE I
     F LOWER$(d$)="n" THEN RETURN           [73A8]
1070 IF LOWER$(d$)<>"j" THEN 1060           [E8F6]
1080 l1=l1-halt:FOR k=1 TO halt:l1=l1+1:
     LOCATE#2,19,l1:PRINT#2,b2alt$(k):NE
     XT                                     [400A]
1090 l=l-h:FOR k=1 TO h:l=l+1:LOCATE#1,1
     9,l:PRINT#1,b2$(k):NEXT                [DD98]
1100 CALL &BB06                             [F05C]
1110 RETURN                                 [8486]
1112 '*****************************         [A8F0]
1114 '    Datensicherung                    [0D44]
1116 '*****************************         [B0F8]
1120 CLS:LOCATE 9,1:PRINT CHR$(24)"*** D
     atensicherung ***"CHR$(24):LOCATE 1
     ,7:PRINT"Bitte Kassette mit der Dat
     ei<2>>>KONTO<< einlegen !!!!"          [7438]
1125 LOCATE 1,12:PRINT"Bitte die Tasten
     "CHR$(24)"Record"CHR$(24)","CHR$(24
     )"Play"CHR$(24)" und "CHR$(24)"ande
     re Taste"CHR$(24)" druecken !!":CAL
     L &BB06                                [9AFC]
1130 SPEED WRITE 1:OPENOUT"!konto":PRINT
     #9,a$                                  [8A34]
1140 PRINT#9,b$                             [8E02]
1150 PRINT#9,b1$                            [0066]
1160 PRINT#9,b4$                            [7E6E]
1170 PRINT#9,b,bs,f,h,i,j,halt,ialt,balt
     ,falt                                  [A8FE]
1180 FOR f1=1 TO f:PRINT#9,c$(f1)           [31FA]
1190 PRINT#9,c(f1),cs(f1),d(f1),e(f1),e1
     (f1):NEXT                              [5692]
1200 FOR h1=1 TO h:PRINT#9,e$(h1)           [2EFC]
1210 PRINT#9,b2$(h1)                        [AA36]
1220 PRINT#9,g$(h1):NEXT                    [FBD0]
1230 FOR f1alt=1 TO falt:PRINT#9,calt$(f
     1alt)                                  [1DFA]
1240 PRINT#9,calt(f1alt):NEXT               [0884]
1250 FOR h1alt=1 TO halt:PRINT#9,ealt$(h
     1alt)                                  [140E]
1260 PRINT#9,galt$(h1alt)                   [B0EA]
1270 PRINT#9,b2alt$(h1alt)                  [5246]
1280 NEXT:CLOSEOUT:GOTO 440                 [3CFA]
1282 '*****************************         [C000]
1284 '    Daten einlesen                    [C5AA]
1286 '*****************************         [7808]
1290 CLS:LOCATE 1,7:PRINT"Bitte Kassette
      mit der Datei<2>>>KONTO<< einlegen
      !!!!":LOCATE 1,12:PRINT"Bitte die
     Taste "CHR$(24)"Play"CHR$(24)" und
     "CHR$(24)"andere Taste"CHR$(24)"dru
     ecken !!":CALL &BB06                   [785E]
1300 OPENIN"!konto":INPUT#9,a1$             [2B8A]
1310 IF a$<>a1$ THEN CLS:CLEAR:INK 2,1,2
     4:PEN 2:LOCATE 11,10:PRINT"FALSCHE
     DATEI !!!!":PEN 1::FOR y=1 TO 10:FO
     R x=1 TO 500:NEXT:PRINT CHR$(7):NEX
     T:GOTO 410                             [A2E2]
1320 INPUT#9,b$                             [8408]
1330 INPUT#9,b1$                            [E96C]
1340 INPUT#9,b4$                            [1374]
1350 INPUT#9,b,bs,f,h,i,j,halt,ialt,balt
     ,falt                                  [D704]
1360 FOR f1=1 TO f:INPUT#9,c$(f1)           [0800]
1370 INPUT#9,c(f1),cs(f1),d(f1),e(f1),e1
     (f1):NEXT                              [AA98]
1380 FOR h1=1 TO h:INPUT#9,e$(h1)           [3614]
1390 INPUT#9,b2$(h1)                        [BB4E]
1400 INPUT#9,g$(h1)                         [F6E4]
1410 NEXT                                   [FB4A]
1420 FOR f1alt=1 TO falt:INPUT#9,calt$(f
     1alt)                                  [3402]
1430 INPUT#9,calt(f1alt):NEXT               [888C]
1440 FOR h1alt=1 TO halt:INPUT#9,ealt$(h
     1alt)                                  [F216]
1450 INPUT#9,galt$(h1alt)                   [ECF2]
1460 INPUT#9,b2alt$(h1alt)                  [F24E]
1470 NEXT:CLOSEIN:RETURN                    [D418]
1480 CLS:END                                [FDC0]
1482 '*****************************         [5404]
1484 '    Statistik                         [C170]
1486 '*****************************         [4C0C]
1490 MODE 1:PRINT CHR$(24)"<5>*** Auswah
     lmoeglichkeiten ***<6>"CHR$(24):PRI
     NT:PRINT:PRINT"1. Statistik anzeige
     n":PRINT:PRINT"2. Statistik neu beg
     innen":PRINT:PRINT"3. Hauptmenue"      [21A0]
1500 LOCATE 1,23:PRINT"Auswahl":LOCATE 1
     ,25:INPUT"",a2:IF a2<1 OR a2>3 THEN
      1500                                  [E30C]
1510 ON a2 GOTO 1520,1590,440               [6D6E]
1520 MODE 2:PRINT CHR$(24);:PRINT SPACE$
     (80):LOCATE 1,1:PRINT"<2>Konto-Stat
     istik von "a1$", gestartet am "b4$"
     ":PRINT CHR$(24)                       [FE46]
1530 LOCATE 1,3:PRINT"Gehalt":LOCATE 27,
     3:PRINT USING"#####.## DM";bs:FOR k
     =1 TO 17:IF k=f+1 THEN 1560 ELSE LO
     CATE 1,k+4:PRINT c$(k):LOCATE 27,k+
     4:PRINT USING"#####.## DM";cs(k):NE
     XT                                     [B3C2]
1540 FOR k=2 TO 22:LOCATE 40,k:PRINT CHR
     $(145):NEXT                            [E7D6]
1550 FOR k=18 TO 34:IF k=f+1 THEN 1560 E
     LSE LOCATE 43,k-13:PRINT c$(k):LOCA
     TE 70,k-13:PRINT USING"#####.## DM"
     ;cs(k):NEXT                            [7382]
1560 FOR k=2 TO 22:LOCATE 40,k:PRINT CHR
     $(145):NEXT                            [ABDA]
1570 WINDOW#4,1,80,23,25:PRINT#4,CHR$(24
     ):CLS#4:LOCATE#4,2,2:PRINT#4,"In di
     eser Statistik werden nur die fest
     angelegten Buchungssaetze verwendet
      !!":PRINT#4,CHR$(24)                  [D54E]
1580 CALL &BB06:GOTO 1490                   [3436]
1590 MODE 1:PRINT"Soll Statistik neu beg
     onnen werden ??":PRINT:PRINT"(J/N)"
     :PRINT:INPUT d$:PRINT                  [77D6]
1600 IF LOWER$(d$)="n" THEN 1490 ELSE bs
     =0:FOR k=1 TO f:cs(k)=0:NEXT:PRINT"
     Statistik<2>ist<2>g e l o e s c h t
     ":b4$=b3$:PRINT:PRINT"Statistik neu
      gestartet am "b4$""                   [EA88]
1610 PRINT:PRINT:PRINT"Weiter mit Taste"
     :CALL &BB06:GOTO 1490                  [E052]
1612 '*****************************         [A2FA]
1614 '    Vorschau                          [3F7C]
1616 '*****************************         [AB02]
1620 l=4:MODE 2:PRINT CHR$(24)TAB(29)"Vo
     rschau ueber 3 Monate"SPACE$(29)CHR
     $(24)                                  [828C]
1630 PLOT 0,373:DRAWR 203,0:DRAWR 0,-312
     :DRAWR -203,0:DRAWR 0,312:PLOT 218,
     373:DRAWR 203,0:DRAWR 0,-312:DRAWR
     -203,0:DRAWR 0,312:PLOT 436,373:DRA
     WR 203,0:DRAWR 0,-312:DRAWR -203,0:
     DRAWR 0,312                            [59AC]
1640 m=VAL(MID$(b$,4,2))                    [A0B8]
1650 m=m+1:IF m>12 THEN m=m-12              [D9D8]
1660 RESTORE:FOR k=1 TO m:READ f$:NEXT      [58E8]
1670 m=m+1:IF m>12 THEN m=m-12              [ABDC]
1680 RESTORE:FOR k=1 TO m:READ f1$:NEXT     [EE4E]
1690 m=m+1:IF m>12 THEN m=m-12              [EDE0]
1700 RESTORE:FOR k=1 TO m:READ f2$:NEXT     [1342]
1710 LOCATE 8,3:PRINT CHR$(24)" "f$" "CH
     R$(24):LOCATE 36,3:PRINT CHR$(24)"
     "f1$" "CHR$(24):LOCATE 63,3:PRINT C
     HR$(24)" "f2$" "CHR$(24)               [ACD4]
1720 g=0:FOR k=1 TO f:IF e(k)=0 THEN l=l
     +1:g=g+c(k):LOCATE 2,l:PRINT c$(k):
     LOCATE 15,l:PRINT USING"#####.## DM
     ";c(k)                                 [67F8]
1730 n=VAL(MID$(b$,4,2))+1:IF n>12 THEN
     n=n-12                                 [A092]
1740 IF e(k)=n THEN l=l+1:g=g+c(k):LOCAT
     E 2,l:PRINT c$(k):LOCATE 15,l:PRINT
      USING"#####.## DM";c(k)               [0E96]
1750 NEXT:LOCATE 2,21:PRINT"Belastung:":
     LOCATE 15,21:PRINT USING"#####.## D
     M";g                                   [B12E]
1760 l=4                                    [CE96]
1770 g=0:FOR k=1 TO f:IF e(k)=0 THEN l=l
     +1:g=g+c(k):LOCATE 29,l:PRINT c$(k)
     :LOCATE 42,l:PRINT USING"#####.## D
     M";c(k)                                [8674]
1780 n=VAL(MID$(b$,4,2))+2:IF n>12 THEN
     n=n-12                                 [7F9E]
1790 IF e(k)=n THEN l=l+1:g=g+c(k):LOCAT
     E 29,l:PRINT c$(k):LOCATE 42,l:PRIN
     T USING"#####.## DM";c(k)              [F812]
1800 NEXT:LOCATE 29,21:PRINT"Belastung:"
     :LOCATE 42,21:PRINT USING"#####.##
     DM";g                                  [0398]
1810 l=4                                    [078E]
1820 g=0:FOR k=1 TO f:IF e(k)=0 THEN l=l
     +1:g=g+c(k):LOCATE 56,l:PRINT c$(k)
     :LOCATE 69,l:PRINT USING"#####.## D
     M";c(k)                                [487E]
1830 n=VAL(MID$(b$,4,2))+3:IF n>12 THEN
     n=n-12                                 [0A98]
1840 IF e(k)=n THEN l=l+1:g=g+c(k):LOCAT
     E 56,l:PRINT c$(k):LOCATE 69,l:PRIN
     T USING"#####.## DM";c(k)              [DA1C]
1850 NEXT:LOCATE 56,21:PRINT"Belastung:"
     :LOCATE 69,21:PRINT USING"#####.##
     DM";g                                  [5BB4]
1860 WINDOW#4,1,80,23,25:PRINT#4,CHR$(24
     ):CLS#4:LOCATE#4,2,2:PRINT#4,"In di
     eser Vorschau werden nur die fest a
     ngelegten Buchungssaetze verwendet
     !!":PRINT#4,CHR$(24)                   [B968]
1870 CALL &BB06:RETURN                      [15AC]
```

**Listing. Immer auf dem laufenden mit »Giro« (Schluß)**

# Schneider ganz analytisch

**Für arbeitsintensive Berechnungen ist ein Computer besonders geeignet. Legen Sie Papier, Bleistift und Taschenrechner zur Seite. Die nächste Fourier-Analyse macht Ihr Schneider.**

Viele physikalische Gesetze leiten sich aus periodischen Schwingungen ab. Kein Wunder, daß dieses Thema in Schule und Universität ausführlich behandelt wird. Eines der wichtigsten Gesetze dabei ist das Zerlegen beliebiger periodischer Funktionen in einfache Sinus- und Cosinuskurven – die Fourier-Analyse. Für jede periodische Schwingung gilt nämlich, daß sie sich durch die Summe einzelner Sinus- und Cosinusfunktionen (mit unterschiedlichen Amplituden und Perioden) darstellen läßt. Die Schwierigkeit liegt, da das mathematische Verfahren sehr aufwendig ist, im Bestimmen der passenden Werte.

Als erstes muß die zu untersuchende Funktion auf die Periodenlänge von $2\pi$ gedehnt beziehungsweise gestaucht werden. Durch einfache mathematische Manipulationen an der Frequenz ist das problemlos möglich. Beim Zerlegen bestimmen Sie zuerst eine Sinus- und eine Cosinusfunktion mit derselben Periodenlänge, wie sie die Gesamtfunktion hat. Dieses Paar heißt »Grundschwingung«. Im nächsten Schritt wählen Sie eine Sinus- und eine Cosinusfunktion mit der doppelten Periode aus. Die Amplituden der bisher vier Kurven müssen Sie so bestimmen, daß die damit erreichte Summenfunktion der Zielfunktion möglichst »nahe«- kommt. Falls ein Glied dieser vier Summen absolut nicht paßt, dann setzen Sie die Amplitude auf 0 – und schon fällt der Term unter den Tisch. Dieses zweite Funktionspaar hört auf den Namen »1. Oberwelle« oder auch »1. harmonische Oberschwingung«.

## Mehr als mathematische Spielereien

Im dritten Schritt versuchen Sie nun, das Fehlende mit einer Sinus- und/oder Cosinusfunktion der dreifachen Periode der Grundschwingung auszugleichen. Die dabei erhaltenen Teilfunktionen nennt man »2. Oberwelle« oder »2. harmonische Oberschwingung«. Diese Schritte führen Sie so lange fort, bis die Summenfunktion die zu untersuchende Funktion hinreichend genau wiedergibt.

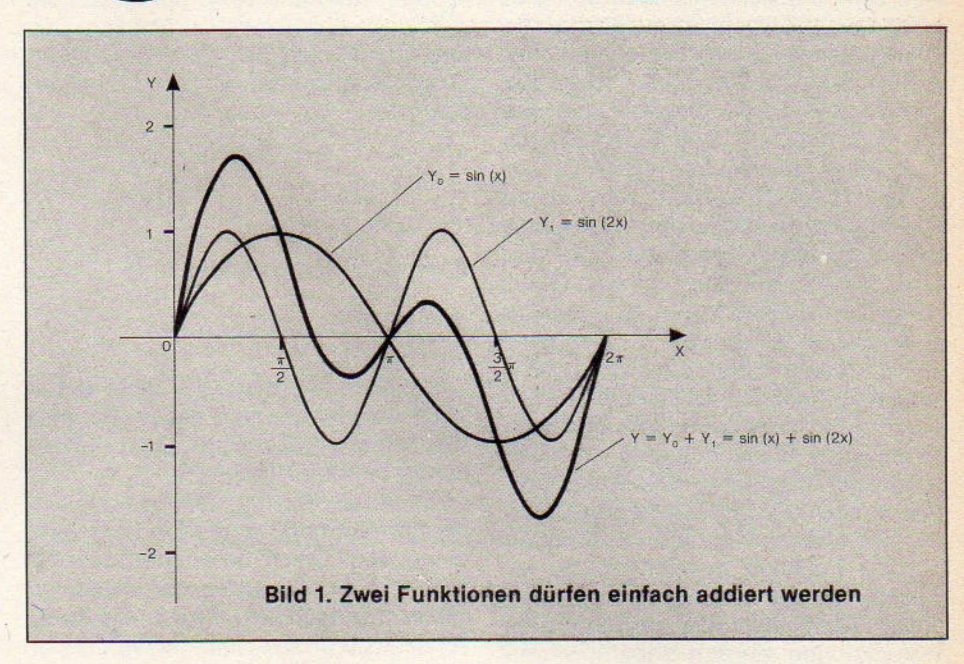



**Bild 1. Zwei Funktionen dürfen einfach addiert werden**

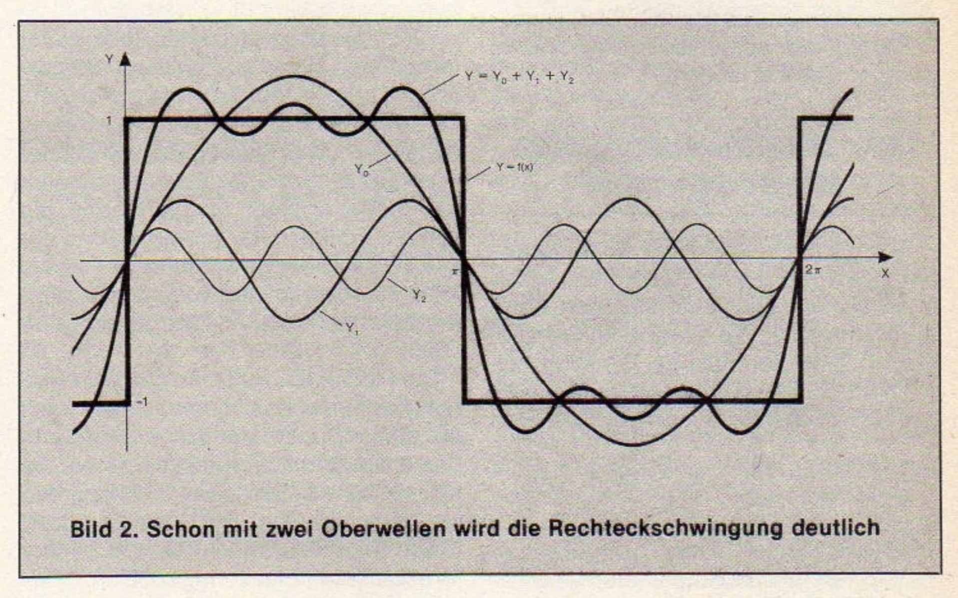



**Bild 2. Schon mit zwei Oberwellen wird die Rechteckschwingung deutlich**

Bild 1 zeigt, wie die Grundfunktion $y_0=\sin(x)$ mit der ersten Oberschwingung $y_1=\sin(2x)$ addiert wird. Variiert man nun die Amplitude (das ist der maximale Ausschlag, den eine Funktion erreichen kann) und benutzt man auch die Cosinusfunktion (diese sieht genauso aus wie die Sinuskurve, erreicht ihr Maximum aber für $x=0$), so kann man schon mit nur einer Oberschwingung verschiedenste Funktionen darstellen. Bild 2 zeigt, wie eine Rechteckkurve mit der Periode $2\pi$ durch Sinusfunktionen dargestellt wird. Sie sehen, daß allein die Grundfunktion mit der 1. und 2. Oberschwingung schon das Charakteristische der Rechteckkurve wiedergibt. Da das Beispiel für $x=0$ den Wert $y(0)=0$ zurückgibt, müssen alle Cosinusglieder die Amplitude 0 bekommen – sie fallen also weg.

Wozu braucht man nun die Fourier-Analyse? In der Elektrotechnik beispielsweise rechnet es sich wesentlich leichter mit sinusförmigen Strömen und Spannungen als mit anderen Signalformen. Im allgemeinen nimmt man deshalb den Mehraufwand, zunächst Strom und Spannungen in Teilfunktionen zu zerlegen, gern in Kauf. Die eigentlichen Berechnungen werden mit den sinusförmigen Teilvorgängen erledigt. Zum Schluß wird der Gesamtvorgang aus den einzelnen Ergebnissen wieder zusammengesetzt.

Auch in der Musik sind Oberwellen von enormer Bedeutung. Jeder Ton ist eine periodische Schwingung, dessen Periodenlänge direkt mit der Tonfre-

quenz zusammenhängt. Je höher ein Ton (und damit die Frequenz), desto kürzer ist die Periodenlänge. Alle Instrumente haben beim gleichen Ton dieselbe Grundschwingung. Einzig die Oberwellen erzeugen die verschiedenen Klängen. Nur sie regeln, ob ein Ton schrill wie eine Trompete oder sanft wie ein Fagott klingt. Somit kann jedes Instrument von elektronischen Orgeln perfekt simuliert werden, wenn diese spezielle Regler für die einzelnen Oberwellen besitzen. Allerdings muß man dazu die Beiträge aller hörbaren Oberwellen herausfinden – ein nahezu unmögliches Vorhaben.

Das Programm aus dem Listing analysiert jede Funktion, die Sie eingeben. Als Ergebnis erhalten Sie Periodenlänge und Amplituden der harmonischen Schwingungen. Die Werte werden grafisch und als Zahlen ausgegeben; die errechnete und die zu untersuchende Funktion gezeichnet. So erkennen Sie auf einen Blick die Abweichung. Die Amplituden der beteiligten Teilfunktionen werden separat veranschaulicht, so daß Sie die Größenverhältnisse der verschiedenen Schwingungen leicht erfassen.

## Der mathematische Hintergrund

Bevor wir uns allerdings den Feinheiten des Programms zuwenden, brauchen wir ein paar Informationen über die Mathematik, die hinter dem Ganzen steckt. Eine periodische Funktion mit n- harmonischen Oberschwingungen läßt sich in folgender Form darstellen:

$$f(x) = \frac{C(0)}{2} + \sum_{k=1}^{n} (a_k \cdot \cos(k \cdot x) + b_k \cdot \sin(k \cdot x))$$

Der griechische Buchstabe »Σ« ist in der Mathematik das Summenzeichen und bedeutet, daß der Ausdruck in den Klammern n-mal addiert wird, wobei für »k« bei jedem Durchgang der nächsthöhere Wert aus dem Bereich zwischen 1 und n eingesetzt wird.

Die Zeichen »$a_k$« und »$b_k$« heißen »Fourier-Koeffizienten«. Sie stellen die Amplitude der Sinus- beziehungsweise Cosinusfunktion dar. Zum Berechnen dieser Koeffizenten dienen die beiden Formeln:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cdot \cos(k \cdot x)\, dx \quad ;$$

$$k = 0, 1, 2, ..., n$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cdot \sin(k \cdot x)\, dx \quad ;$$

$$k = 1, 2, 3, ..., n$$

Sie brauchen übrigens nicht unbedingt verstehen, wie diese Formeln funktionieren. Wichtig ist allein, daß sie richtig benutzt werden.

Für »k« werden nacheinander die hinter den Formeln angegebenen Werte eingesetzt. Damit berechnen sich jeweils die Koeffizienten der einzelnen Oberschwingungen. »f(x)« ist die Funktion, die durch die Sinus- und Cosinuskurven ausgedrückt werden soll.

»∫« ist das Integralzeichen und bedeutet, daß mit Hilfe der Funktion, die zwischen dem Integralzeichen und dem Term »dx« steht, die Fläche zwischen der Kurve und der x-Achse berechnet wird. Grafisch ist damit der Koeffizient »$a_k$« nichts anderes als die Fläche, die von der Funktion $f(x) * \cos(k * x)$ und der x-Achse im Bereich zwischen $x = -\pi$ bis $x = \pi$ eingeschlossen wird. Der schraffierte Bereich von Bild 3 zeigt solch eine Fläche.

Die Fläche der Funktion kann man näherungsweise bestimmen, indem man nur einige Punkte der Kurve berechnet und diese durch Geraden verbindet. Je kleiner der Abstand zweier Hilfspunkte ist, desto genauer wird das Ergebnis. Diese neue Fläche läßt sich in trapezförmige Teilstücke zerlegen. Deren Fläche wird mit der Formel

Fläche des Trapez =

$\frac{1}{2}$ (Grundlinie + Decklinie) · Höhe

berechnet (siehe Bild 4). Bild 5 zeigt, wie die Funktion in einzelne Trapeze aufgeteilt wird. Falls man n+1-Punkte der Funktion berechnet und diese gleichmäßig über den gesamten Bereich verteilt, so hat jedes Trapez die Breite »$h = 2 * \pi / n$«. Die Fläche eines Trapezes unserer Funktion berechnet sich dann mit

$$A_i = \frac{1}{2} \cdot h \cdot (f(x_i) \cdot \cos(k \cdot x_i) + f(x_{i+1}) \cdot \cos(k \cdot x_{i+1}))$$

oder vereinfacht mit

$$A_i = \frac{\pi}{n} (f(x_i) \cdot \cos(k \cdot x_i) + f(x_{i+1}) \cdot \cos(k \cdot x_{i+1}))$$

Die gesamte Fläche und damit das gesamte Integral berechnet sich als Summe aller n-Trapeze

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (f(x_i) \cdot \cos(k \cdot x_i) + f(x_{i+1}) \cdot \cos(k \cdot x_{i+1}))$$

oder ausgeschrieben

$$A = \frac{2}{n} \left( \left( \frac{\varphi_0}{2} + \frac{\varphi_1}{2} \right) + \left( \frac{\varphi_1}{2} + \frac{\varphi_2}{2} \right) + \left( \frac{\varphi_2}{2} + \frac{\varphi_3}{2} \right) + ... \right)$$

mit

$$\varphi_i = f(x_i) \cdot \cos(k \cdot x_i)$$

Durch Verschieben der Klammern ändert sich der Ausdruck in

$$A = \frac{2}{n} \left( \frac{\varphi_0}{2} + \left( \frac{\varphi_1}{2} + \frac{\varphi_1}{2} \right) + \left( \frac{\varphi_2}{2} + \frac{\varphi_2}{2} \right) + ... \right)$$

und zusammengefaßt

$$A = \frac{2}{n} \left( \frac{\varphi_0}{2} + \sum_{i=1}^{n-1} \varphi_i + \frac{\varphi_n}{2} \right)$$

Da wir hier nur periodische Funktion betrachten, gilt speziell in diesem Fall $\varphi_0 = \varphi_n$. Statt »$\frac{\varphi_0}{2} + \frac{\varphi_n}{2}$« kann man also auch $\frac{\varphi_0}{2} + \frac{\varphi_0}{2}$ schreiben und damit die zwei störenden Glieder mit in die Summe aufnehmen. Die gesamte Formel für die Integralfläche lautet also:

$$A = \frac{2}{n} \sum_{i=0}^{n-1} \varphi_i$$

In dem hier besprochenen speziellen Fall ist $\varphi$ das Produkt aus einer Cosinus-

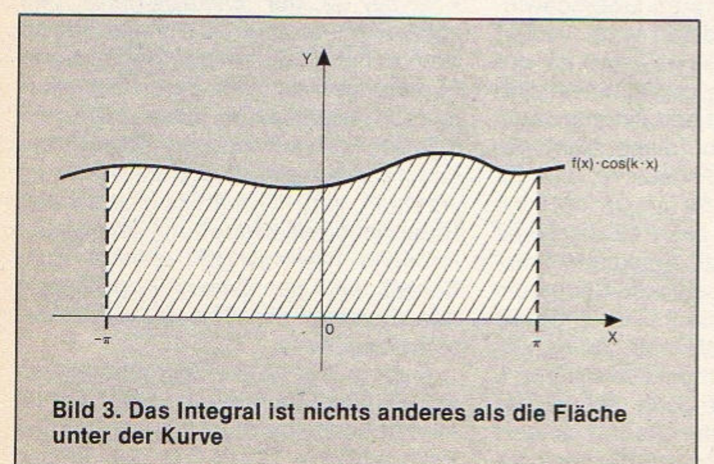



**Bild 3. Das Integral ist nichts anderes als die Fläche unter der Kurve**

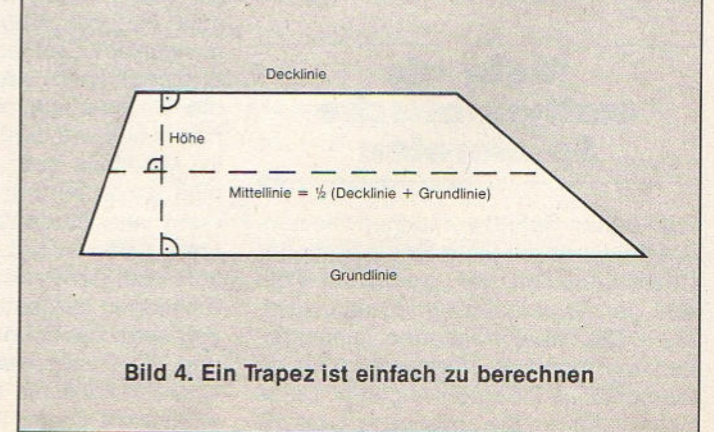



**Bild 4. Ein Trapez ist einfach zu berechnen**

Funktion (bei dem zweiten Koeffizienten einer Sinus-Funktion) und einer beliebigen Funktion f(x). »f(x)« ist dabei die Funktion, die durch die Fourier-Koeffizienten angenähert werden soll. Zu berechnen sind also

$$a_k = \frac{2}{n} \sum_{i=1}^{n} (f(x_i) \cdot \cos(k \cdot x_i))$$

und

$$b_k = \frac{2}{n} \sum_{i=1}^{n} (f(x_i) \cdot \sin(k \cdot x_i))$$

Wie oben schon erwähnt, muß man, je genauer die Integralflächen bestimmt werden sollen, mehr Punkte – und damit Teilflächen – berechnen. Störend wirken sich dabei die vielen trigonometrischen Funktionen aus, die in der Rechenzeit ganz gewaltig zu Buche schlagen. Aber die Mathematiker haben auch für dieses Problem eine Lösung: »Goertzels Algorithmus«.

Dieser Algorithmus besagt, daß die beiden Summen

$$a_k = \sum_{k=0}^{n} \varphi_k \cdot \cos(k\xi)$$

$$b_k = \sum_{k=0}^{n} \varphi_k \cdot \sin(k\xi)$$

für unsere Koeffizienten nach folgender Regel berechnet werden:

1. Als Anfangsbedingung wählt man

$U_{n+1} = U_{n+2} = 0$
und
$t = 2 \cdot \cos(\xi)$

2. Danach berechnet man nacheinander für

$j = n, n\text{-}1, n\text{-}2, ..., 2, 1$
$U_j = \varrho_j + t \cdot U_{j+1} - U_{j+2}$

3. Das Ergebnis (die Cosinus- und die Sinussumme) berechnet sich durch

$a_k = \varrho_0 + U_1 \cdot \cos(\xi) - U_2$
$b_k = U_1 \cdot \sin(\xi)$

Sie brauchen jetzt für beliebige Mengen von Funktionswerten nur noch einmal den Cosinus berechnen. Auch ansonsten reduziert sich der Aufwand innerhalb der später zu programmierenden Schleife beträchtlich.

Das einzige Problem, das wir jetzt noch haben, ist die richtige Wahl der Anfangsbedingungen. Wenn man jedoch die zu berechnende Gleichung und die Ausgangsformel des »Goertzels Algorithmus« untereinander schreibt, erkennt man die Lösung auf den ersten Blick:

$$a_k = \frac{2}{n} \sum_{i=1}^{n} f_i \cdot \cos(k \cdot \frac{2 \cdot \pi}{n} \cdot i)$$

$$b_k = \frac{2}{n} \sum_{i=1}^{n} f_i \cdot \sin(k \cdot \frac{2 \cdot \pi}{n} \cdot i)$$

und

$$a_k = \sum_{k=0}^{n} \varrho_k \cdot \cos(k \cdot \xi)$$

$$b_k = \sum_{k=0}^{n} \varrho_k \cdot \sin(k \cdot \xi)$$

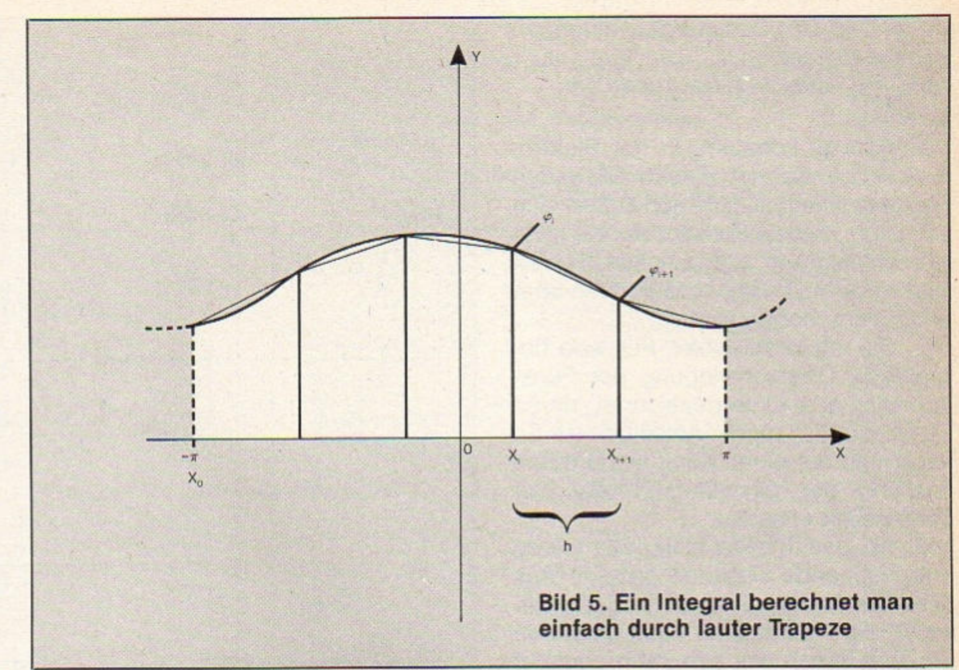



**Bild 5. Ein Integral berechnet man einfach durch lauter Trapeze**

Statt »ξ« setzen Sie $\xi = k * 2 * \pi/2$ ein. Zusätzlich müssen die Summen nach dem Berechnen mit dem Wert 2/n multipliziert werden, um die richtigen Integrale und damit die endgültigen Größen der Fourier-Koeffizienten zu erhalten.

## Das Hauptprogramm zur Fourier-Analyse

Jetzt haben wir alle Kunstgriffe kennengelernt, um unser Programm optimal zu schreiben. Durch Vertauschen der Variablen bei jedem Schleifendurchlauf kommt man statt den n+3-Variablen $U_0$ bis $U_{n+2}$ mit insgesamt drei Variablen aus: mit U0, U1 und U2. Die daraus folgende Programmfassung des Goertzels Algorithmus finden Sie ab der Zeile 11200.

Beim Übergang von einer harmonischen Oberschwingung zur nächsten greifen wir zu einem weiteren Trick: Das Argument »q« der Sinus- und Cosinusberechnung erhöht sich bei jedem Übergang von einer beliebigen harmonischen Oberschwingung zur nächsten gleichmäßig um den Wert $2 * \pi/n$. Somit lassen sich die neuen Sinus- und Cosinuswerte nach den folgenden Formeln berechnen:

cos(q+d) :=
cos(q)*cos(d)−sin(q)*sin(d)

sin(q+d) :=
cos(q)*sin(d)+sin(q)*cos(d)

Da die aktuellen Werte cos(q) und sin(q) bekannt sind, gehen in die Berechnung der neuen Werte nur der Sinus und Cosinus der Schrittweite ein. Die Schrittweite wiederum ist aber während der ganzen Fourier-Analyse konstant. Man darf deren Bestimmung also aus der Schleife herausziehen und braucht sie im Programm nur ein einziges Mal zu berechnen.

In dem ganzen Programm muß somit der Sinus und der Cosinus nur einmal berechnet werden. Überhaupt sind in dem Programm alle öfter benötigten Werte möglichst aus den Schleifen herausgezogen. Falls Sie Ihre Programme immer auf solche Faktoren überprüfen, bekommen Sie häufig einen ungeahnten Geschwindigkeitszuwachs. Insbesondere ist es sinnvoll, die n-Funktionswerte in einer Tabelle fertig ausgerechnet an das Analyseprogramm zu übergeben. Auch bei sehr umfangreichen Analysen muß so jeder Funktionswert nur ein einziges Mal ermittelt werden.

Das Programm teilt sich in mehrere logische Einheiten. Im Eingabeteil werden alle Parameter bestimmt. Danach wird im Vorbereitungsteil Ihre Eingabe in das interne Format umgerechnet. Es werden beispielsweise die Funktionswerte berechnet und in die Tabelle eingetragen. Nach der Analyse wird das Ergebnis »nachbereitet«. Es werden die maximalen und minimalen Werte gesucht und eine Skalierung vorbereitet. Der Ausgabeteil zeichnet die Bilder auf den Bildschirm. Unabhängig von den absoluten Funktionswerten füllt eine Funktion genau den zur Verfügung stehenden Raum aus.

Ausgegeben werden fünf verschiedene Darstellungen der bearbeiteten Funktion:

1. Die Funktion selbst, so wie sie eingegeben wurde.
2. Eine Tabelle mit den Zahlenwerten der einzelnen Fourier-Koeffizienten. Diese Werte können bei anderen Berechnungen weiterverwendet werden. Der Wert der 0. Oberschwingung hat im Sinusglied keine Bedeutung, da sin(0)

immer 0 ist. Das Cosinusglied hingegen ist sehr wichtig:

$c_0(x)=a_0*\cos(0*x)=a_0*\cos(0)=a_0*1=a_0$

Der Wert ist konstant. In der Elektrotechnik bezeichnet dieses Glied den Gleichstromanteil. Grafisch äußert sich das darin, daß die Funktionswerte nicht gleichmäßig um die x-Achse verteilt sind, sondern im Mittel nach oben oder unten verschoben sind.

3. Die »Spektrallinien«: Für jede berechnete Oberschwingung der Funktion wird eine Linie gezeichnet, deren Länge ein Maß für die Amplitude ist. So kann man auf einen Blick die Größenordnung der einzelnen Anteile der Oberwellen erfassen.

4. Als gestrichelte Linie wird die zu untersuchende Funktion gezeigt. Aus der Anzahl der Punkte beziehungsweise der »Dichte« der Linie erkennen Sie, wie genau das Integral berechnet wurde. So erfassen Sie auf einen Blick, ob das berechnete Ergebnis überhaupt einen Sinn hat. In der Regel sollten mindestens doppelt so viele Punkte berechnet werden, wie harmonische Oberschwingungen bearbeitet wurden.

5. Mit einer durchgezogenen Linie wird die durch die Fourier-Analyse erhaltene Näherung angezeigt. Diese Anzeige überlagert im selben Maßstab die gestrichelte Linie, so daß Sie sofort die Größe des Restfehlers abschätzen können. Eine Bildschirmausgabe finden Sie in Bild 6.

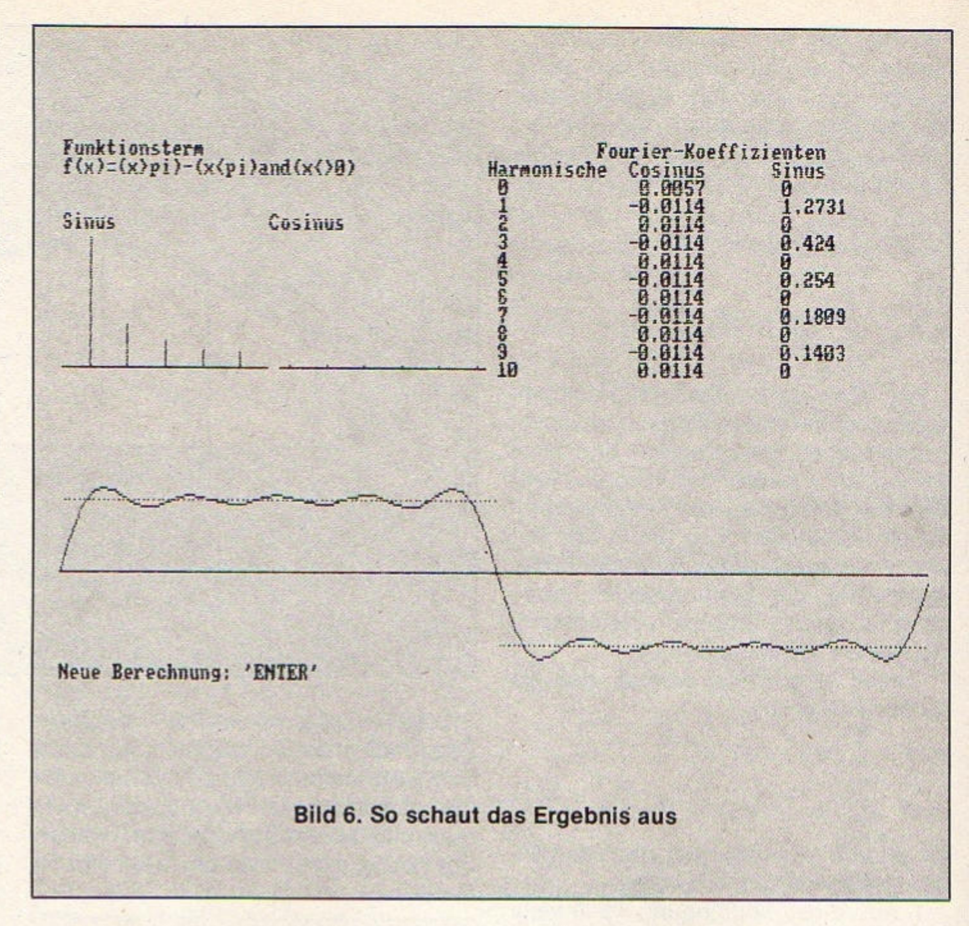



**Bild 6. So schaut das Ergebnis aus**

Der Eingabeteil ist der »unmathematischste« Teil des Gesamtprogramms, gleichzeitig aber auch der trickreichste. Ein besonderes »Schmankerl« ist die Eingabe der zu untersuchenden Funktion. Diese muß nämlich irgendwie in der Zeile 10530 landen, damit deren Wert in die Berechnung eingeht. Auf eine »saubere« Art und Weise ist das mit dem Locomotive-Basic nicht möglich. In dem Programm wird die Funktion zunächst ganz normal mit »INPUT« in eine Zeichenkette eingelesen. Danach wird eine Funktionstaste (die kleine <ENTER>-Taste) mit einer Zeichenkette, bestehend aus der Zeilennummer, in der die Funktion stehen soll, dem Text der Funktion, dem Drücken der <ENTER>-Taste, einer »GOTO-Fortsetzungszeile« und einem zweiten Druck auf <Enter> belegt. Jetzt wird die Schreibfarbe auf 0 gesetzt (um verräterische Bildschirmausgaben zu vermeiden) und das Programm abgebrochen.

Der Computer gibt jetzt die Meldung »Ready« aus, was aber wegen der Schreibstiftfarbe unsichtbar bleibt. Für Sie schaut es so aus, als ob Ihr Schneider während eines Programmlaufs auf eine Eingabe wartet. Drücken Sie jetzt die vorbereitete <ENTER>-Taste, so wird (immer noch unsichtbar) in einem Rutsch die Funktionszeile ins Programm eingefügt und dieses erneut gestartet. Erst jetzt wird die Bildschirmausgabe wieder auf »sichtbar« geschaltet.

Die eingegebene Funktion wird dabei allerdings nicht auf Syntaxfehler überprüft. Falls Sie eine unsinnige Funktion eingeben, bricht das – ansonsten korrekte Programm – mit »Syntax Error in Line 10530« ab. Sie können dann allerdings die Zeile korrigieren und das Programm mit »CONT« fortsetzen – oder ohne Korrektur mit »RUN« neu starten.

Obwohl im Menü maximal eine 8-fache Genauigkeit angefordert wird, dürfen Sie problemlos größere Werte eingeben. Sie müssen aber selbst entscheiden, ob die ansteigende Rechenzeit durch die höhere Genauigkeit aufgewogen wird. Sinnvolle Ergebnisse bringt das Programm nur ab »einfacher Genauigkeit«. Klar ist auch, daß die Rechenzeit mit wachsender Zahl der Oberschwingungen ansteigt.

Funktionen, die in verschiedenen x-Abschnitten verschieden definiert sind, können Sie mit dieser Eingaberoutine nur sehr umständlich eingeben. Eine Rechteckfunktion beispielsweise hat die Form:

$f(x)=(x>\pi)-(x<\pi)$ AND $x<>0)$

Im Bereich 0 bis $\pi$ ist die Aussage »$x>\pi$« falsch und »$x<\pi$« wahr. Bei dem CPC hat »unwahr« den Wert »0« und »wahr« den Wert »–1«. In diesem Bereich hat die Gesamtfunktion also den Wert 1 (=0–(–1)). Falls x größer als $\pi$ ist, ist »$x>\pi$« wahr und »$x<\pi$« falsch. Die Funktion hat dann also den Wert –1 (=(–1)–0).

Durch »AND ($x<>0$)« gilt das Ganze allerdings nur, solange x ungleich Null ist. Für x=0 wird »$x<>0$« falsch und die Gesamtfunktion erhält an dieser Stelle den Wert 0.

Einen einfachen Test, ob das Programm richtig arbeitet, unternehmen Sie, wenn Sie die oben beschriebene Rechteckfunktion mit sechs harmonischen Oberschwingungen und doppelter Genauigkeit testen. Jetzt müssen nur die Sinuswerte der geradzahligen harmonischen Oberschwingungen einen Wert ungleich Null besitzen. Bei anderen Werten können aber kleine Fehler auftreten. Diese kommen zustande, wenn die Abtastpunkte für das Integral nicht »symmetrisch« um die Symmetriepunkte der Funktion verteilt sind. Bei der Rechteckfunktion tritt dieser Effekt sowohl bei anderen Genauigkeiten als auch bei anderen Zahlen der Oberwellen auf. (Helmut Tischer/hg)

| Steckbrief | |
|---|---|
| Name: | Fourier-Analyse |
| Computer | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger | Diskette/Kassette |





```
10000 '***********************************
      ***********************************
      *****                              [1010]
10010 '*          F O U R I E R - A N A
      L Y S E              (Vers.  2.02.
      86) *                              [FE00]
10020 '* (c) 23.07.85 by Isar-Amper-Soft

         *                               [F2D8]
10030 '***********************************
      ***********************************
      *****                              [9516]
10040 '                                  [3878]
10050 DIM f(0),c(0),s(0)'Dummy-Vorbelegu
      ngen                               [4DB4]
10060 MODE 2:INK 0,1:INK 1,24:BORDER 1:P
      APER 0:PEN 1                       [D31E]
10070 GOSUB 10180'Eingabe                [F608]
10080 GOTO 10370'Berechnung vorbereiten  [76B4]
10090 GOSUB 11100'Berechnung ausfuehren  [4156]
10100 GOSUB 10590'Berechnung nachbereite
      n                                  [F9C2]
10110 GOSUB 10710'Ausgabe                [4F16]
10120 LOCATE 1,25:PRINT"Neue Berechnung:
       'ENTER'                           [20DA]
10130 q$=INKEY$                          [C516]
10140 IF q$=CHR$(13) THEN GOTO 10060     [6518]
10160 GOTO 10130                         [49CC]
10170 '                                  [1580]
10180 'Funktionseingabe                  [1F9A]
10190 PRINT TAB(20)"F O U R I E R - A N
      A L Y S E"                         [B758]
10200 PRINT TAB(20)"====================
      =========="                        [58EE]
10210 PRINT                              [FE42]
10220 PRINT"Zu analysierende Funktion (
      Im Bereich 0 bis 2*PI periodisch):
      "                                  [A30E]
10230 PRINT                              [F846]
10240 INPUT"f(x)=",f$                    [751A]
10250 PRINT                              [1A4A]
10260 INPUT"Wieviele Harmonische sollen
      berechnet werden";h%               [3A44]
10270 PRINT                              [744E]
10280 PRINT"<3>(0) : halbe<5>Genauigkeit
      "                                  [32A0]
10290 PRINT"<3>(1) : einfache<2>Genauigk
      eit"                               [9652]
10300 PRINT"<3>(2) : doppelte<2>Genauigk
      eit"                               [7698]
10310 PRINT"<3>(3) : vierfache Genauigke
      it"                                [47FC]
10320 PRINT"<3>(4) : 8-fache<3>Genauigke
      it"                                [1FDE]
10330 INPUT"Mit welcher Genauigkeit soll
       gerechnet werden";g%              [0F8E]
10340 PRINT                              [754A]
10350 RETURN                             [D8F2]
10360 '                                  [1482]
10370 'Fourier-Analyse vorbereiten       [59BA]
10380 '                                  [3A86]
10390 PRINT                              [2A54]
10400 PRINT TAB(10)"Bitte jetzt die 'ENT
      ER'-Taste im Funktionsblock drueck
      en,"                               [3786]
10410 PRINT TAB(20)"Um die Berechnung zu
       starten"                          [E8EC]
10420 PEN 0                              [6394]
10430 KEY 139,CHR$(13)+"10530 f(i%)="+f$
      +CHR$(13)+"GOTO 10450"+CHR$(13)    [33C6]
10440 END'Warten auf Tastendruck         [E718]
10450 PEN 1:KEY 139,CHR$(13)             [0920]
10460 PRINT TAB(20)"Bitte warten - Berec
      hnung laeuft"                      [EA9C]
10470 '                                  [1586]
10480 n%=(h%+1)*2^g%'Anzahl der Abtastst
      ellen                              [7086]
10490 ERASE f,c,s:DIM f(n%),c(n%),s(n%)  [DD6C]
10500 fmax=0:fmin=0                      [D400]
10510 FOR i%=0 TO n%-1'Funktionswerte an
       den Abtaststellen berechnen       [EF04]
10520 x=2*PI*i%/n%                       [7F78]
10530 f(i%)=SIN(x)                       [BB9C]
10540 fmax=MAX(f(i%),fmax):fmin=MIN(f(i%
      ),fmin)                            [AA90]
10550 NEXT                               [22B4]
10560 f(n%)=f(0)                         [4314]
10570 GOTO 10090                         [E7E0]
10580 '                                  [168A]
10590 'Ergebnisse nachbereiten           [47AA]
10600 cmax=0:smax=0:cmin=0:smin=0        [D972]
10610 FOR i%=0 TO h%                     [3A14]
10620 cmax=MAX(c(i%),cmax):smax=MAX(s(i%
      ),smax)                            [B7D6]
10630 cmin=MIN(c(i%),cmin):smin=MIN(s(i%
      ),smin)                            [04C0]
10640 NEXT                               [A7B4]
10650 scale=128/(fmax-fmin)              [5EFE]
10660 ybase=-fmin*scale+32               [900E]
10670 scale2=115/(MAX(cmax,smax)-MIN(cmi
      n,smin))                           [92BA]
10680 ybase2=200-MIN(cmin,smin)*scale2   [9F08]
10690 RETURN                             [4A00]
10700 '                                  [3A7E]
10710 'Ausgabe der Ergebnisse            [54F4]
10720 CLS                                [B8F8]
10730 '                                  [3984]
10740 'Angabe der Koeffizienten          [C59A]
10750 PRINT TAB(50)"Fourier-Koeffiziente
      n"                                 [55EA]
10760 ZONE 13:PRINT TAB(40)"Harmonische"
      ,"Cosinus","Sinus"                 [D7B8]
10770 FOR i%=0 TO h%:PRINT TAB(40)i%,ROU
      ND(c(i%),4),ROUND(s(i%),4):NEXT    [BAD2]
10780 'Angabe der Funktion               [169C]
10790 LOCATE 1,1:PRINT"Funktionsterm"    [2616]
10800 WINDOW#7,1,39,2,5:PRINT#7,"f(x)="+
      f$                                 [5062]
10810 '                                  [4382]
10820 'Summenfunktion                    [454A]
10830 MOVE 639,ybase:DRAW 0,ybase        [BC9A]
10840 y=0:FOR j%=0 TO h%:y=y+c(j%):NEXT:
      MOVE 0,ybase+y*scale               [E4AE]
10850 FOR i%=0 TO 640 STEP 8             [5BA2]
10860 x=2*PI*i%/640:cosx=COS(x):sinx=SIN
      (x)                                [8530]
10870 '2 mal der Goertzels Algorithmus f
      uer Cosinus bzw. Sinus-Anteil      [5CDE]
10880 u1=0:u2=0:v1=0:v2=0:t=2*cosx       [7246]
10890 FOR j%=h% TO 1 STEP -1             [8BE0]
10900 u0=c(j%)+t*u1-u2:u2=u1:u1=u0:v0=s(
      j%)+t*v1-v2:v2=v1:v1=v0            [E48E]
10910 NEXT                               [26B4]
10920 y=c(0)+u1*cosx-u2+v1*sinx          [D400]
10930 DRAW i%,ybase+y*scale              [401E]
10940 NEXT                               [B2BA]
10950 MOVE 0,ybase+f(0)*scale            [6E54]
10960 FOR i%=0 TO n%                     [5330]
10970 PLOT 640*i%/n%,ybase+scale*f(i%)   [C4EC]
10980 NEXT                               [0AC2]
10990 'Spektrallinien                    [A61E]
11000 LOCATE 1,5:PRINT"Sinus"TAB(20)"Cos
      inus"                              [2496]
11010 MOVE 0,ybase2:DRAWR 151,0:MOVER 9,
      0:DRAWR 151,0                      [443C]
11020 st=152/(h%+1)                      [5472]
11030 FOR i%=0 TO h%                     [FA0E]
11040 MOVE (i%+0.5)*st,ybase2:DRAWR 0,sc
      ale2*s(i%):NEXT                    [0CE4]
11050 FOR i%=0 TO h%:MOVE (i%+0.5)*st+16
      0,ybase2:DRAWR 0,scale2*c(i%):NEXT
                                         [97A2]
11060 RETURN                             [D5F0]
11070 '                                  [3480]
11080 '                                  [3982]
11090 '                                  [3A84]
11100 'Fourier-Analyse, Hauptprogramm    [3286]
11110 '==============================    [A0C2]
11120 'Eingaben:                         [AF1E]
11130 ' h%: Anzahl der Harmonischen      [E3F8]
11140 ' n%: Anzahl der Abtastpunkte (dab
      ei sollte n%>2*h%+1 sein)          [9F70]
11150 ' f(0) bis f(n%-1): Funktionswerte
       (von 0 bis 2*pi)                  [9DEA]
11160 ' DIM c(h%),s(h%): 2 auf mindesten
      s h% dimensionierte Felder c und h
                                         [3BD4]
11170 'Ausgaben:                         [9542]
11180 ' c(0)           : Amplitude  der K
      onstanten                          [1E8A]
11190 ' c(1) bis c(h%): Amplituden der C
      osinus-Harmonischen                [87BE]
11200 ' s(1) bis s(h%): Amplituden der S
      inus-Harmonischen                  [CB4A]
11210 '                                  [3678]
11220 d=2/n%:sind=SIN(d*PI):cosd=COS(d*P
      I):sinq=0:cosq=1                   [81DE]
11230 'q hat in Abh{ngigkeit der Harmoni
      schen den Wert q = k%*d*pi = k%*2*
      pi/n%                              [9A96]
11240 FOR k%=0 TO h%                     [2618]
11250 'Goertzels Algorithmus: Summen ueb
      er f(i)*sin(i*q) und f(i)*cos(i*q)
                                         [AA48]
11260 u1=0:u2=0:t=2*cosq                 [CEF0]
11270 FOR i%=n%-1 TO 1 STEP -1:u0=f(i%)+
      t*u1-u2:u2=u1:u1=u0:NEXT           [52DE]
11280 c(k%)=d* (f(0)+u1*cosq-u2)         [4A10]
11290 s(k%)=d* (u1*sinq)                 [5870]
11300 'Sinus/Cosinuswerte fuer q:=q+d*pi
       --> q:=k%*2*pi/n% berechnen       [9340]
11310 neu=cosq*cosd-sinq*sind:sinq=cosq*
      sind+sinq*cosd:cosq=neu            [62A4]
11320 NEXT                               [A4AC]
11330 c(0)=c(0)/2                        [8AFC]
11340 RETURN                             [D8F2]
11350 '                                  [3682]
11360 END                                [CFE4]
```

**Listing. Fourier-Analyse per Computer**

# Der neue »CPC«

**Ein Bombenerfolg war unser Maschinencode-Prüfsummer »CPC« im letzten Schneider-Sonderheft. Was den Commodore-64-Besitzern ihr »MSE«, ist den Schneider-Besitzern seitdem der »CPC«. Jetzt ist er sowohl kürzer als auch noch leistungsfähiger geworden.**

Zur Eingabe langer Maschinencode-Programme ist nichts empfehlenswerter als das Programm »CPC«. Der Name kommt nicht von ungefähr – er steht für »Comfortable Program for Codeinput«. Diesen Komfort gewährleisten die vielfältigen Fähigkeiten des CPC. Er ist praktisch an beliebige Maschinencode-Listings anzupassen: Weder vor DATA-Ladern noch vor Hexdumps streckt »CPC« seine Waffen. Einzige Bedingung für Basic-Lader mit Prüfsummen ist, daß die Zahl der Byte pro Zeile konstant bleibt. Vielen brauchen wir die Details sicher nicht zu wiederholen, da sie bereits aus dem letzten Beitrag bekannt sind. Für neu hinzugekommene Leser führen wir sie hier dennoch auf. Die Vorzüge der neuen Version:

- Dateinamen kann man jetzt in beliebigem Format eingeben. Wer mit Kassetten arbeitet, darf also Namen mit einer Länge von bis zu 16 Zeichen verwenden, wer Disketten bevorzugt, setzt wahlweise die Laufwerksnummer oder den User-Bereich voran.
- »CPC« setzt HIMEM automatisch auf Adresse 20000.
- Der Aufruf der Routine »Parameter einstellen« erfolgt automatisch bei der Code-Eingabe oder DATA-Erzeugung.
- Das Unterprogramm »Parametereingabe« arbeitet komfortabler und ist dadurch einfacher einzustellen. Die einzelnen Punkte sind numeriert. Ein Druck der jeweiligen Zifferntaste wechselt die Einstellungen (zum Beispiel von »hex« auf »dez«), wie Sie in Bild 1 sehen können.
- Die Routine »Erzeuge DATAs« verarbeitet jetzt auch die Startadresse 8000 hex korrekt und die Vorgabe eines Offset ist berichtigt.
- Der erzeugte Basic-Lader ist kürzer und schneller und erhält automatisch den SAVE-Befehl zur Speicherung des erzeugten Maschinencodes.
- Jetzt sind auch DATA-Lader ohne Prüfsumme oder mit Add- statt mit Hash-Prüfsumme zu erzeugen.
- Durch diverse Einsparungen hat »CPC« nur noch eine Länge von 10 KByte.
- Zwei Hilfsprogramme unterstützen das Hauptprogramm »CPC.BAS« (Listing 2). Für noch komfortablere Bedienung belegt »CPC.HLP« (Listing 1) die Funktionstasten mit allen Hex-Ziffern und setzt die Farben auf eine augenfreundliche Kombination. »CPC.INF« finden Sie nur auf der Leserservice-Diskette, denn es enthält eine Kurzanleitung für »CPC«.
- Die Routine »Code eingeben« verarbeitet jetzt bis zu 128 Byte formatierter Eingabe. So sind nun auch DATA-Lader mit sehr langen Zeilen abzutippen.
- Die Eingabe von Dezimalzahlen war bisher etwas kompliziert. Um das nötige dreistellige Format einzuhalten, waren bei ein- oder zweistelligen Zahlen entweder führende Nullen voranzuschicken oder stets die Leertaste (oder <ENTER>) zu drücken. Nun ist, wenn Sie im Parameter-Menü »Ende annehmen« auf »Nein« einstellen, jedes Byte mit der Leertaste oder <ENTER>, beziehungsweise dem Dezimalpunkt zu bestätigen.

Das Programm »CPC« verhilft in komfortabler Weise zu einer einfachen, schnellen und sicheren Eingabe von Maschinencode-Programmen. »CPC«-Benutzer geben nur zirka 60 Prozent des Listingumfangs von Basic-Ladern ein.

Nach dem Start erscheint das Hauptmenü mit fünf Punkten. Die Eingabe der Anfangsbuchstaben ruft das jeweilige Unterprogramm auf.

**Lade Code**

lädt eine Binärdatei von Kassette oder Diskette. Die Ladeadresse des Programms müssen Sie eingeben; sie darf jedoch nicht unter 20000 liegen. Andernfalls laden Sie es an eine höhere Adresse. Drücken Sie bei der Aufforderung, den Datenträger bereitzumachen, <ESC> oder <CTRL+C>, bricht die Routine ab.

**Schreibe Code**

sichert den Inhalt eines Speicherbereichs auf Diskette oder Kassette. Sie müssen die Anfangs- und die Endadresse angeben. Beide Werte lassen sich an den Zeilenadressen der DATA-Listings ablesen. Haben Sie den Code in einen anderen Bereich geladen oder eingegeben, weil er sonst unterhalb der Adresse 2000 stände, berücksichtigt CPC das hier nicht. Sie müssen später den Code an die richtige Adresse laden (zum Beispiel: »LOAD "CODE",3000«). Auch diese Routine ist mit <ESC> oder <CTRL+C> abzubrechen.

**Code eingeben**

ist die wichtigste Routine des »CPC«. Zunächst geben Sie im automatisch erscheinenden Unter-Menü »Parameter einstellen« die für das Listing erforderlichen Standards ein. Darauf folgt die Vorgabe der ersten Zeilennummer und der Schrittweite.

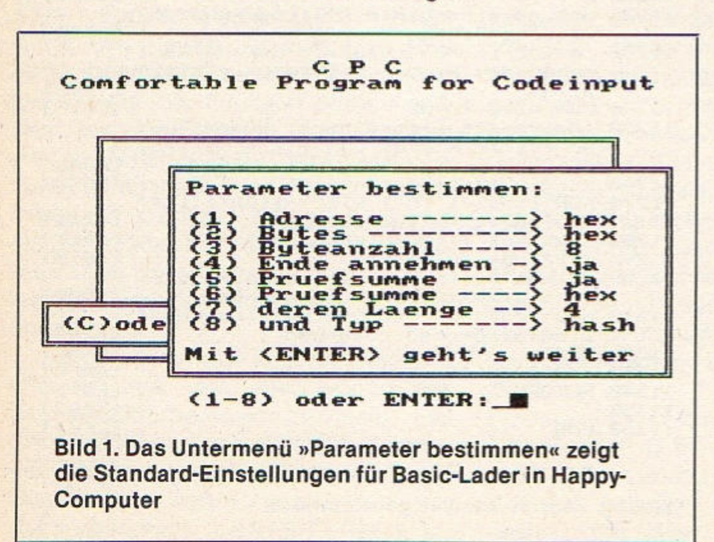



Bild 1. Das Untermenü »Parameter bestimmen« zeigt die Standard-Einstellungen für Basic-Lader in Happy-Computer

```
1000 '--> CPC.HLP -- Voreinstellungen       [A3EC]
1010 '                                      [9212]
1020 '--> Farben                            [EC20]
1030 '                                      [9016]
1040 BORDER 4:INK 0,4:INK 1,26:INK 3,18     [0A90]
1050 '                                      [B61A]
1060 '--> Tastatur - Zehnerblock            [B004]
1070 '                                      [B41E]
1080 KEY DEF 15,1,48,127,128                [A192]
1090 KEY DEF 13,1,49,47,129                 [B936]
1100 KEY DEF 14,1,50,47,130                 [7408]
1110 KEY DEF 5,1,51,47,131                  [2EAE]
1120 KEY DEF 20,1,52,65,132                 [150E]
1130 KEY DEF 12,1,53,66,133                 [C718]
1140 KEY DEF 4,1,54,67,134                  [7CC2]
1150 KEY DEF 10,1,55,68,135                 [CD24]
1160 KEY DEF 11,1,56,69,136                 [9C2E]
1170 KEY DEF 3,1,57,70,137                  [81C6]
1180 KEY DEF 7,1,46,38,138                  [6DD6]
1190 KEY DEF 6,0,13,139,140                 [D01E]
1200 '                                      [9514]
1210 'Version vom 22.10.86                  [BA24]
```

Listing 1. So ist der Zehnerblock hilfreich belegt

Drücken Sie hier einfach <ENTER>, bleibt der Zeilenzähler auf Null. Nun geben Sie noch die Startadresse vor. Bei der Eingabe der Byte-Werte ist die Gefahr von Fehleingaben sehr gering, da nur die Ziffern 0 bis 9, beziehungsweise die Buchstaben A bis F zugelassen sind. Ein Druck der Leertaste, <ENTER> oder des Punktes formatiert alle führenden Nullen automatisch und schließt die Eingabe eines Byte ab. Die Tasten mit Schrägstrichen (</> und <\>) wiederholen das zuletzt eingegebene Byte. Dadurch ist gewährleistet, daß Sie nur ein Minimum einzugeben haben. Verzichten Sie auf Prüfsummen (wovon wir aber eindringlich abraten, denn etwaige Eingabefehler sind so kaum zu finden), ist damit die Eingabe beendet. Ansonsten geben Sie nun die Prüfsummen ein. Ist die Prüfsumme korrekt, ertönt ein Signalton und Sie gehen zur nächsten Zeile. Fehler korrigieren Sie mit Hilfe der DEL-Taste.

```
100 '*************************************  [31D4]
101 '*BEISPIEL.DAT - DATA-Lader von 'CPC'*  [1BA8]
102 '*************************************  [A3D8]
103 '                                       [DEB6]
104 DATA 9C40,01,0A,A0,21,0E,A0,CD,D1,17AB [489C]
105 DATA 9C48,BC,C9,12,A0,18,0A,00,00,64E8 [3F74]
106 DATA 9C50,00,00,53,43,41,4C,C5,00,0CE2 [0616]
107 '                                       [DEBE]
108 '                                       [FEC0]
109 '                                       [FEC2]
110 '                                       [DDB2]
126 DATA 9CF0,19,22,A8,AC,22,2C,B3,2A,1B2C [CBCA]
127 DATA 9CF8,AA,AC,22,2E,B3,E1,23,C1,7E3B [3820]
128 DATA 9D00,10,93,FB,C9,00,00,00,00,3F30 [A316]
129 DATA *ENDE*                            [83CC]
130 adr=&9C40:zeile=104:MEMORY adr-1      [EA36]
131 READ d$:IF d$="*ENDE*"THEN 142        [0182]
132 '                                       [DFBA]
133 '                                       [DFBC]
134 '                                       [FFBE]
135 '                                       [FFC0]
140 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                  [3E0A]
141 zeile=zeile+1:GOTO 131                 [B052]
142 SAVE"BEISPIEL.BIN",B,&9C40,&C7:END     [0AF6]
```

**Bild 2. Ein beispielhafter Ausschnitt eines typischen DATA-Laders. Die Ziffern im unterlegten Bereich sind in jedem Fall einzugeben, während Sie auf die Eingabe der Prüfsummen (eingerahmt) verzichten können. Dazu kommen noch die Startadresse (in diesem Fall 9C40 hex) und zum Speichern die Endadresse (hier 9D03 hex). Den Rest des Basic-Laders ersparen Sie sich mit »CPC«.**

**Erzeuge DATAs**

erzeugt aus Maschinencode im Arbeitsspeicher einen lauffähigen Basic-Lader auf Diskette. Gehen Sie bitte wie folgt vor: Stellen Sie zuerst die korrekten Parameter ein. Wählen Sie dann die Namen für den DATA-Lader und die später vom Lader zu erzeugende Binär-Datei. Nun erwartet »CPC« die Anfangs- und die Endadresse des Maschinencodes. Da dieser nicht unter 20000 beginnen darf, manche Programme aber nur auf niedrigeren Adressen arbeiten, können Sie hier einen Offset von der Ladeadresse zur tatsächlichen Startadresse eingeben.

Beispielsweise steht ein Programm im Speicher ab Adresse 6000 hex, soll aber so gespeichert werden, daß der Basic-Lader es auf 4000 hex erzeugt. Die Eingaben sind für diesen Fall:

`Startadresse=&4000, Offset=&2000.`

Wenn Sie kein Offset benötigen, drücken Sie einfach <ENTER>. Jetzt fehlt nur noch die Nummer der ersten Zeile und die Schrittweite der Numerierung, bevor »CPC« mit der Erzeugung des Laders beginnt und ihn als ASCII-Datei auf Kassette oder Diskette speichert.

Das Menü »Parameter bestimmen« bricht man mit <ESC> oder <CTRL+C> ab und beendet es mit <ENTER>. Die Tasten 1 bis 8 ändern einzelne Parameter:

<1> Zeilenadresse dezimal oder hexadezimal anzeigen.

<2> Byte dezimal oder hexadezimal erwarten und anzeigen.

<3> Anzahl der Byte pro Zeile. Eingabe in Form einer Zahl kleiner oder gleich 128.

<4> Ende annehmen ist normalerweise »Ja«. Haben Sie »zwei Ziffern« bei hexadezimaler oder »drei Ziffern« bei dezimaler Eingabe gewählt, geht »CPC« automatisch zum nächsten Byte über. Ist dieser Punkt »Nein«, müssen Sie jedes Byte mit der Leertaste, <ENTER> oder dem Punkt beenden.

<5> Prüfsumme abfragen. »Nein« ist nur bei Basic-Ladern ohne Prüfsumme zu empfehlen.

<6> Prüfsumme dezimal oder hexadezimal erwarten und anzeigen.

<7> Länge der Prüfsumme. Minimal 4 bei hexadezimaler und 5 bei dezimaler Ausgabe (Voreinstellung). »CPC« benötigt diese Angabe hauptsächlich für das Erzeugen der Basic-Lader.

<8> Prüfsummentyp. Viele Basic-Lader verwenden Prüfsummen des Typs »Add«, also eine einfache Addition aller Bytes einer Zeile. Die »CPC«-Lader verfügen jedoch über eine Hash-Prüfsumme, die Fehler und Vertauschungen erkennt. Falscheingaben sind hier fast unmöglich. Die Parameter des in Happy-Computer verwendeten Eingabeformats sehen Sie in Bild 1.

Während der Arbeit mit »CPC« beenden Sie jede Eingabe mit <ENTER> und korrigieren mit <DEL>. Für Dateinamen müssen Diskettenbenutzer ein gültiges Format wählen. Bei allen Zahleneingaben ist eine dezimale oder hexadezimale Eingabe mit vorangestelltem »$«, »#« oder »&« wählbar. An jeder Stelle, an der eine Taste zur Bestätigung zu drücken ist, läßt sich die jeweilige Funktion durch <ESC> oder <CTRL+C> abbrechen. (Stefan Aust/ja)

| Steckbrief | |
|---|---|
| Programm: | CPC |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

```
1000 '                                        [9110]
1010 '--> "CPC = Comfortable Program for
     Codeinput" by St. M. Aust                [53B0]
1020 '                                        [8F14]
1030 '                                        [9016]
1040 '--> Initialisierung                     [CFE8]
1050 '                                        [B61A]
1060 CLOSEIN:MEMORY 19999                     [AC78]
1070 KEY DEF 66,0,3:CALL &BB48                [9D26]
1080 brk$=CHR$(3):bell$=CHR$(7):back$=CH
     R$(8):cr$=CHR$(13):clr$=CHR$(16)         [D184]
1090 del$=back$+clr$:cur1$=CHR$(143)+bac
     k$:cur2$=CHR$(211)+back$                 [A504]
1100 dz$="0123456789":hx$="0123456789ABC
     DEF"                                     [0BA6]
1110 DIM b$(128),fl(8):fl(1)=1:fl(2)=1:f
     l(3)=8:fl(5)=1:fl(6)=1:fl(7)=4:fl(8
     )=1                                      [52BA]
1120 ENV 1,15,-1,20:ENV 2,15,-1,4             [2772]
1130 '                                        [9018]
1140 '--> Menue                               [BA7E]
1150 '                                        [921C]
1160 MODE 1:PAPER 0:PEN 1:PAPER#1,0:PEN#
     1,1:PAPER#2,0:PEN#2,3                    [3464]
1170 LOCATE 18,1:PRINT"C P C"                 [7450]
1180 LOCATE 4,2:PRINT"Comfortable Progra
     m for Codeinput"                         [01EE]
1190 LOCATE 6,7:PEN 1:PRINT"Geschrieben
     von Stefan M.Aust"                       [B9C0]
1200 LOCATE 8,8:PRINT"Version II - Oktob
     er 1986"                                 [7E6A]
1210 SOUND 1,400,0,15,1:SOUND 2,450,0,15
     ,1:SOUND 4,500,0,0,1                     [C3B6]
1220 WHILE SQ(1)<>4:WEND                      [B074]
1230 x=6:y=6:xl=29:yl=15:GOSUB 3310:WIND
     OW SWAP 1                                [0EBC]
1240 LOCATE 9,3:PRINT"(L)ade Code"            [A788]
1250 LOCATE 7,5:PRINT"(S)chreibe Code"        [F2E8]
1260 LOCATE 7,7:PRINT"(C)ode eingeben"        [E11E]
1270 LOCATE 7,9:PRINT"(E)rzeuge DATAs"        [267C]
1280 LOCATE 9,11:PRINT"(B)eende CPC"          [B93E]
1290 WINDOW#2,1,40,23,23:PRINT#2,TAB(12)
     "Bitte waehlen: ";cur1$;                 [F4D2]
1300 GOSUB 3410:p=INSTR("LSCEB"+brk$,a$)
     :IF p=0 THEN PRINT bell$;:GOTO 1300
                                              [566E]
1310 x=3:y=17:xl=18:yl=3:GOSUB 3310           [2432]
1320 PRINT#2,a$:ON p GOTO 1330,1420,1520
     ,1880,2670,2670                          [7CF4]
1330 '                                        [901C]
1340 '--> Lade Code                           [4BB0]
1350 '                                        [B620]
1360 PRINT#1,"<2>(L)ade Code"                 [EDBA]
1370 WINDOW SWAP 2:PRINT TAB(10)"Name: "
     ;:GOSUB 2860:n$=b$                       [5CE6]
1380 PRINT TAB(11)"Startadresse: ";:GOSU
     B 2750:start=b                           [8240]
1390 PRINT"Disk/Kassette einlegen & Tast
     e druecken"                              [7CF6]
1400 GOSUB 3410:IF brk THEN 1130              [FC52]
1410 LOAD"!"+n$,start:GOTO 1130               [4EB4]
1420 '                                        [971C]
1430 '--> Schreibe Code                       [7F0E]
1440 '                                        [9520]
1450 PRINT#1,"(S)chreibe Code"                [3798]
1460 WINDOW SWAP 2:PRINT TAB(10)"Name: "
     ;:GOSUB 2860:n$=b$                       [13E6]
1470 PRINT TAB(11)"Startadresse: ";:GOSU
     B 2750:start=b                           [0F40]
1480 PRINT TAB(12)"Endadresse: ";:GOSUB
     2750:finish=b                            [D5FC]
1490 PRINT"Disk/Kassette einlegen & Tast
     e druecken"                              [01F8]
1500 GOSUB 3410:IF brk THEN 1130              [DA54]
1510 SAVE"!"+n$,b,start,finish-start:GOT
     O 1130                                   [0D00]
1520 '                                        [B91E]
1530 '--> Code eingeben                       [CB40]
1540 '                                        [9322]
1550 PRINT#1,"(C)ode eingeben"                [48CA]
1560 GOSUB 2470:IF brk THEN 1130 ELSE WI
     NDOW SWAP 2                              [1A46]
1570 PRINT TAB(11)"Startnummer: ";:GOSUB
      2750:nr=b                               [BC00]
1580 PRINT TAB(10)"Schrittweite: ";:GOSU
     B 2750:inc=b                             [B96E]
1590 PRINT TAB(10)"Startadresse: ";:GOSU
     B 2750:start=b                           [6044]
1600 GOSUB 3230:LOCATE#1,32,2:PRINT#1,"(
     C)ode eingeben"                          [D770]
1610 IF fl(2)THEN h$="&":l=2 ELSE h$="":
     l=3                                      [7F1E]
1620 h=l+1:xa=16:xe=xa+h*15                   [37BA]
1630 PRINT:PRINT USING"(#####) ";nr;:IF
     fl(1)THEN PRINT" "HEX$(start,4)": "
     ;ELSE PRINT USING"#####: ";start;        [A692]
1640 FOR i=0 TO fl(3):b$(i)="":NEXT           [BAEA]
1650 p=0:x=xa                                 [1A22]
1660 y=VPOS(#0):IF x<xa THEN x=xe:y=y-1
     ELSE IF x>xe THEN PRINT CHR$(10)CHR
     $(11):x=xa:y=VPOS(#0)                    [DD3E]
1670 b$=b$(p):LOCATE x,y:PRINT b$;            [5B9E]
1680 lmax=l:IF fl(2)THEN GOSUB 2950 ELSE
      GOSUB 2970                              [0644]
1690 IF fl(2)=0 THEN b=VAL(h$+b$):IF b>2
     55 THEN PRINT bell$;:GOTO 1680           [7202]
1700 IF brk THEN 1130 ELSE IF del=0 THEN
      1730                                    [AEFC]
1710 b$(p)="":IF p>0 THEN p=p-1:x=x-h EL
     SE del=0:PRINT bell$;                    [049C]
1720 GOTO 1660                                [0A20]
1730 LOCATE x,y:PRINT b$;:b$(p)=b$            [0698]
1740 x=x+h:p=p+1:IF p<fl(3)THEN 1660          [6ECC]
1750 FOR i=0 TO fl(3)-1:POKE start+i,VAL
     (h$+b$(i)):NEXT                          [24F8]
1760 IF fl(5)=0 THEN 1830                     [EA3A]
1770 GOSUB 3480:LOCATE x,y:PRINT"= ";:x=
     x+2:b$="":lmax=fl(7)                     [A670]
1780 IF fl(6)THEN GOSUB 2950 ELSE GOSUB
     2970                                     [5124]
1790 IF brk THEN 1130 ELSE IF del THEN x
     =x-2:PRINT del$;del$;:GOTO 1710          [AD8C]
1800 LOCATE x,y:PRINT b$;                     [A10C]
1810 IF fl(6)THEN pr2=VAL("&"+b$)ELSE pr
     2=VAL(b$)                                [766A]
1820 IF pr<>pr2 THEN PRINT bell$;:GOTO 1
     780                                      [CE0E]
1830 start=start+fl(3):nr=nr+inc              [2C44]
1840 SOUND 1,400,0,15,2                       [B1A6]
1850 GOTO 1630                                [2422]
1860 b$(p)="":IF p>1 THEN p=p-1:x=x-h EL
     SE del=0:PRINT bell$;                    [A0AA]
1870 GOTO 1660                                [752C]
1880 '                                        [9930]
1890 '--> Erzeuge DATA's                      [7CF8]
1900 '                                        [BF22]
1910 PRINT#1,"(E)rzeuge DATAs"                [B322]
1920 GOSUB 2470:IF brk THEN 1130              [BC6A]
1930 x=2:y=7:xl=37:yl=9:GOSUB 3310:WINDO
     W SWAP 1                                 [A468]
1940 LOCATE 2,1:PRINT"DATA-File-Name: ";
     :GOSUB 2860:dn$=b$                       [680E]
1950 LOCATE 2,2:PRINT"Code-File-Name: ";
     :GOSUB 2860:cn$=b$                       [D2D2]
1960 LOCATE 4,4:PRINT"Startadresse: ";:G
     OSUB 2750:start=b                        [BC40]
1970 LOCATE 6,5:PRINT"Endadresse: ";:GOS
     UB 2750:finish=b                         [CC00]
1980 LOCATE 10,7:PRINT"Offset: ";:GOSUB
     2750:offs=b                              [FE88]
1990 x=2:y=17:xl=37:yl=4:GOSUB 3310:WIND
     OW SWAP 1                                [86CC]
2000 LOCATE 4,1:PRINT"Erste Nummer: ";:G
     OSUB 2750:nr=b                           [7DC6]
2010 LOCATE 4,2:PRINT"Schrittweite: ";:G
     OSUB 2750:inc=b                          [B34E]
2020 PRINT#2,"Disk/Kassette einlegen & T
     aste druecken"                           [48E6]
2030 GOSUB 3410:IF brk THEN 1130              [FE52]
2040 form=1:IF form THEN nn$="00000"ELSE
      nn$=""                                  [B09E]
2050 st2=start:start=start+offs:finish=f
     inish+offs:nr2=nr+4*inc                  [712A]
2060 d1=6:IF fl(5)THEN d1=d1+5+(fl(8)=0)
                                              [EC16]
2070 d2=5:IF fl(5)THEN d2=d2+5+(fl(8)=0)
                                              [341C]
2080 PRINT#2,"Erzeugung beginnt...";          [8888]
2090 OPENOUT"!"+dn$                           [E436]
2100 h$="* "+dn$+" - DATA-Lader von 'CPC
     ' *"                                     [13CE]
2110 PRINT#9,nr;"'"+STRING$(LEN(h$),42):
     nr=nr+inc                                [A4C0]
2120 PRINT#9,nr;"'"+h$:nr=nr+inc              [92A6]
2130 PRINT#9,nr;"'"+STRING$(LEN(h$),42):
     nr=nr+inc                                [48C4]
2140 PRINT#9,nr;"'":nr=nr+inc                 [473C]
2150 PRINT#9,nr;"DATA ";:nr=nr+inc            [37DA]
2160 IF fl(1)THEN PRINT#9,HEX$(start-off
     s,4);ELSE PRINT#9,USING"#####";star
     t-offs;                                  [8548]
2170 FOR i=0 TO fl(3)-1:IF fl(2)THEN IF
     form THEN PRINT#9,",";HEX$(PEEK(sta
     rt+i),2);:GOTO 2190 ELSE PRINT#9,",
     ";HEX$(PEEK(start+i));:GOTO 2190         [33B8]
2180 a$=STR$(PEEK(start+i)):PRINT#9,",";
     RIGHT$(nn$+MID$(a$,2),3);                [206E]
2190 NEXT:IF fl(5)=0 THEN PRINT#9:GOTO 2
     220                                      [9514]
2200 GOSUB 3480:IF fl(6)THEN PRINT#9,","
     ;HEX$(pr,fl(7)):GOTO 2220                [8F7E]
2210 a$=STR$(pr):PRINT#9,",";RIGHT$(nn$+
     MID$(a$,2),fl(7))                        [958E]
2220 IF INKEY$=brk$THEN CALL &BC92:GOTO
     1130                                     [94A4]
2230 start=start+fl(3):IF start<finish T
     HEN 2150                                 [7318]
2240 PRINT#9,nr;"DATA *ENDE*":nr=nr+inc       [4C44]
2250 PRINT#9,nr;"adr=";:IF fl(1)THEN PRI
     NT#9,"&"+HEX$(st2,4);ELSE PRINT#9,M
     ID$(STR$(st2),2);                        [513A]
2260 PRINT#9,":zeile="+MID$(STR$(nr2),2)
```

```
     ;:nr=nr+inc                                [CB04]
2270 PRINT#9,":MEMORY ";:IF st2=32768 TH
     EN PRINT#9,"&7FFF"ELSE PRINT#9,"adr
     -1"                                        [9752]
2280 PRINT#9,nr;"READ d$:IF d$="+CHR$(34
     )+"*ENDE*"+CHR$(34)+"THEN";nr+d1*in
     c:nr=nr+inc                                [61E4]
2290 IF fl(5)THEN PRINT#9,nr;"pr=0":nr=n
     r+inc                                      [2544]
2300 PRINT#9,nr;"FOR i=1TO";fl(3):nr=nr+
     inc                                        [E40E]
2310 IF fl(2)THEN PRINT#9,nr;"READ a$:a=
     VAL("+CHR$(34)+"&"+CHR$(34)+"+a$)":
     nr=nr+inc ELSE PRINT#9,nr;"READ a":
     nr=nr+inc                                  [E450]
2320 PRINT#9,nr;"POKE adr,a:adr=adr+1":n
     r=nr+inc                                   [C496]
2330 IF fl(5)=0 THEN 2370 ELSE IF fl(8)T
     HEN 2350                                   [A146]
2340 PRINT#9,nr;"pr=pr+a":nr=nr+inc:GOTO
      2370                                      [BACA]
2350 PRINT#9,nr;"pr=pr*2:IF pr>65535THEN
      pr=pr-65535":nr=nr+inc                    [C5DA]
2360 PRINT#9,nr;"pr=UNT(pr)XOR a:if pr<0
     THEN pr=pr+65536":nr=nr+inc                [6B72]
2370 PRINT#9,nr;"NEXT i":nr=nr+inc              [3088]
2380 IF fl(5)=0 THEN 2420                       [7830]
2390 IF fl(6)THEN PRINT#9,nr;"READ pr$:p
     r2=VAL("+CHR$(34)+"&"+CHR$(34)+"+pr
     $):if pr2<0then pr2=pr2+65536":nr=n
     r+inc:GOTO 2410                            [E97C]
2400 PRINT#9,nr;"READ pr2":nr=nr+inc            [1A8C]
2410 PRINT#9,nr;"IF pr<>pr2 THEN PRINT"+
     CHR$(34)+"Pruefsummenfehler in Zeil
     e"+CHR$(34)+";zeile:STOP":nr=nr+inc
                                                [9F52]
2420 PRINT#9,nr;"zeile=zeile+";MID$(STR$
     (inc),2);":GOTO";nr-d2*inc:nr=nr+in
     c                                          [6BAC]
2430 PRINT#9,nr;"SAVE"+CHR$(34)+cn$+CHR$
     (34)+",B,&"+HEX$(st2)+",&"+HEX$(fin
     ish-st2):nr=nr+inc                         [C3D4]
2440 PRINT#9,nr;"PRINT d$:END":CLOSEOUT         [BC98]
2450 PRINT#2," und ist fertig.";bell$           [70C2]
2460 GOSUB 3410:GOTO 1130                       [4658]
2470 '                                          [B728]
2480 '--> Parameter einstellen                  [3B82]
2490 '                                          [B92C]
2500 x=10:y=8:xl=28:yl=14:GOSUB 3310            [5098]
2510 WINDOW SWAP 0,1:PRINT"Parameter bes
     timmen:"                                   [2A24]
2520 LOCATE 1,3:SOUND 1,900,0,15,2              [DCEC]
2530 PRINT"(1) Adresse -------> ";:IF fl
     (1)THEN PRINT"hex"ELSE PRINT"dez"          [AA2E]
2540 PRINT"(2) Bytes ---------> ";:IF fl
     (2)THEN PRINT"hex"ELSE PRINT"dez"          [6968]
2550 PRINT"(3) Byteanzahl ---->";fl(3)          [4B2C]
2560 PRINT"(4) Ende annehmen -> ";:IF fl
     (4)THEN PRINT"nein"ELSE PRINT"ja<2>
     "                                          [B2BC]
2570 PRINT"(5) Pruefsumme ----> ";:IF fl
     (5)THEN PRINT"ja<2>"ELSE PRINT"nein
     "                                          [3456]
2580 PRINT"(6) Pruefsumme ----> ";:IF fl
     (6)THEN PRINT"hex"ELSE PRINT"dez"          [7B02]
2590 PRINT"(7) deren Laenge -->";fl(7)          [6D60]
2600 PRINT"(8) und Typ -------> ";:IF fl
     (8)THEN PRINT"hash"ELSE PRINT"add "
                                                [BDCA]
2610 PRINT:PRINT"Mit <ENTER> geht's weit
     er"                                        [FAAA]
2620 PRINT#2,TAB(11)"(1-8) oder ENTER:_"
     ;cur1$;                                    [9A86]
2630 GOSUB 3410:IF a$=brk$OR a$=cr$ THEN
      RETURN                                    [7E58]
2640 IF a$<"1"OR a$>"8"THEN PRINT bell$;
     :GOTO 2630 ELSE PRINT#2,a$:f=VAL(a$
     )                                          [5276]
2650 IF f<>3 AND f<>7 THEN fl(f)=1-fl(f)
     :GOTO 2520                                 [49A2]
2660 WINDOW SWAP 0,2:PRINT TAB(12)"neuer
      Wert: ";:GOSUB 2750:fl(f)=b:WINDOW
      SWAP 0,2:GOTO 2520                        [6DF8]
2670 '                                          [972C]
2680 '--> Beende CPC                            [FD10]
2690 '                                          [BD30]
2700 PRINT#1,"<2>(B)eende CPC"                  [D808]
2710 WINDOW SWAP 2:PRINT TAB(9)"Zurueck
     mit ESC oder ^C"                           [01A8]
2720 GOSUB 3410:IF brk THEN 1130                [FD5E]
2730 MODE 2                                     [3EC6]
2740 END                                        [F888]
2750 '                                          [952A]
2760 '--> Sub: Hex-Dez-Input                    [E208]
2770 '                                          [972E]
2780 b$="":b=0:lmax=5                           [C5BE]
2790 GOSUB 3150:IF a=13 AND b$<>"&"THEN
     2840                                       [60C0]
2800 IF a=127 THEN GOSUB 3200:GOTO 2790         [CB14]
2810 IF b=0 THEN IF a>34 AND a<39 THEN i
     n$=hx$:b$="&":b=1:PRINT a$;:GOTO 27
     90 ELSE in$=dz$                            [2EE8]
2820 IF INSTR(in$,a$)>0 AND b<lmax THEN
     b$=b$+a$:b=b+1:PRINT a$;ELSE PRINT
     bell$;                                     [5426]
2830 GOTO 2790                                  [5630]
2840 b=VAL(b$):IF b<0 THEN b=b+65536            [9D44]
2850 RETURN                                     [839E]
2860 '                                          [C82E]
2870 '--> Sub: Filename-Input                   [3C64]
2880 '                                          [C232]
2890 b$="":b=0:lmax=16                          [E926]
2900 GOSUB 3150:IF a=13 THEN RETURN             [C41C]
2910 IF a=127 THEN GOSUB 3200:GOTO 2900         [700A]
2920 IF a>31 AND a<127 AND b<lmax THEN b
     $=b$+a$:b=b+1:PRINT a$;ELSE PRINT b
     ell$;                                      [0BCC]
2930 GOTO 2900                                  [6324]
2940 '                                          [C82C]
2950 '--> Sub: Hex-Zahl eingeben                [8122]
2960 in$=hx$:GOTO 2990                          [6128]
2970 '--> Sub: Dez-Zahl eingeben                [1F22]
2980 in$=dz$:GOTO 2990                          [1128]
2990 '                                          [C536]
3000 '--> Sub: Input                            [45AC]
3010 '                                          [B816]
3020 b=LEN(b$):brk=0:IF del THEN del=0:G
     OTO 3100 ELSE del=0                        [BF62]
3030 GOSUB 3150                                 [A69E]
3040 IF a=3 THEN brk=1:GOTO 3120                [46B8]
3050 IF a=13 OR a=32 OR a=46 THEN 3120          [9936]
3060 IF a=47 OR a=92 THEN b$=bb$:b=LEN(b
     b$):RETURN                                 [6960]
3070 IF a=127 THEN 3100                         [4708]
3080 IF INSTR(in$,a$)>0 AND b<lmax THEN
     b$=b$+a$:b=b+1:PRINT a$;ELSE PRINT
     bell$;                                     [0624]
3090 IF fl(4)THEN 3030 ELSE IF b=lmax TH
     EN 3120 ELSE 3030                          [59A8]
3100 IF b>0 THEN GOSUB 3200:GOTO 3030           [E120]
3110 PRINT bell$;:del=1                         [B2DA]
3120 b$=RIGHT$("00000"+b$,lmax)                 [74BE]
3130 bb$=b$:RETURN                              [A058]
3140 '                                          [B91E]
3150 '--> Sub: Tastdruck nach a,a$              [80DA]
3160 PRINT cur2$;                               [D0E4]
3170 a$=UPPER$(INKEY$):IF a$=""THEN 3170
                                                [2FB6]
3180 PRINT clr$;                                [E872]
3190 a=ASC(a$):RETURN                           [7BA4]
3200 '--> Sub: DEL-Routine                      [B460]
3210 IF b>0 THEN b=b-1:b$=LEFT$(b$,b):PR
     INT back$;clr$;ELSE PRINT bell$;           [359E]
3220 RETURN                                     [878E]
3230 '                                          [921E]
3240 '--> Sub: Rahmen zeichnen                  [1936]
3250 '                                          [B822]
3260 MODE 2                                     [2BC4]
3270 MOVE 112,352:DRAW 527,352:DRAW 527,
     399:DRAW 112,399:DRAW 112,352              [BD9E]
3280 MOVE 116,356:DRAW 523,356:DRAW 523,
     395:DRAW 116,395:DRAW 116,356              [81B0]
3290 LOCATE 28,25:PRINT"Zurueck mit ESC
     oder ^C"                                   [F34E]
3300 WINDOW 1,80,4,24:RETURN                    [325E]
3310 '                                          [901C]
3320 '--> Sub: Fenster oeffnen                  [8046]
3330 '                                          [9220]
3340 WINDOW#1,x,x+xl-1,y,y+yl-1:CLS#1           [51E6]
3350 xp=x*16-16:yp=415-y*16:xm=xl*16-1:y
     m=yl*16-1                                  [397C]
3360 PLOT xp,yp,1:DRAWR xm,0:DRAWR 0,-ym
     :DRAWR -xm,0:DRAWR 0,ym                    [5FDA]
3370 xp=xp+4:yp=yp-4:xm=xm-8:ym=ym-8            [85A2]
3380 PLOT xp,yp,3:DRAWR xm,0:DRAWR 0,-ym
     :DRAWR -xm,0:DRAWR 0,ym                    [27E2]
3390 WINDOW#1,x+1,x+xl-2,y+1,y+yl-2             [5984]
3400 RETURN                                     [7D8E]
3410 '                                          [C01E]
3420 '--> Sub: Auf Taste warten                 [F074]
3430 '                                          [C222]
3440 WHILE INKEY$<>"":WEND                      [A79C]
3450 a$=UPPER$(INKEY$):IF a$=""THEN 3450
                                                [87BA]
3460 IF a$=brk$THEN brk=1 ELSE brk=0            [50A4]
3470 RETURN                                     [9F9C]
3480 '                                          [972C]
3490 '--> Sub: Checksum bilden                  [0428]
3500 '                                          [BD1E]
3510 pr=0:IF fl(8)THEN 3540                     [8D2E]
3520 FOR i=0 TO fl(3)-1:pr=pr+PEEK(start
     +i):NEXT                                   [49EA]
3530 RETURN                                     [8C96]
3540 FOR i=0 TO fl(3)-1:pr=pr*2:IF pr>65
     535 THEN pr=pr-65535                       [BA5A]
3550 pr=UNT(pr)XOR PEEK(start+i):IF pr<0
      THEN pr=pr+65536                          [F3C4]
3560 NEXT:RETURN                                [2B8E]
```

Listing 2. Eingabe von Maschinencode im Eiltempo

# Backup-Master

**Ein universelles Kopierprogramm, das in alle Richtungen arbeitet, steht bei vielen Computer-Benutzern ganz oben auf dem Wunschzettel. Sie können diese Position jetzt getrost auf Ihrer Liste streichen.**

Daß man von seinen Programmen immer mindestens je eine Sicherheitskopie anfertigen sollte, weiß wohl jeder – spätestens nach ersten üblen Erfahrungen. Doch gab es bislang kein wirklich universelles Kopierprogramm, das seinem Benutzer diese Fleißarbeit abnahm. Das übernimmt jetzt »Backup-Master«, denn es kopiert

- von Kassette auf Kassette
- von Kassette auf Diskette
- von Diskette auf Kassette
- von Diskette auf Diskette.

Als ganz besondere Fähigkeit kommt hinzu, daß er auch headerlose Kassettendateien kopiert und dabei deren Kennbyte sogar automatisch erkennt.

Es lassen sich also ganze Disketten oder Kassetten in einem »Rutsch« wahlweise auf das jeweils andere Speichermedium übertragen. Beim Kopieren auf Kassette erreichen Sie durch das 2-Block-Verfahren und hohe Baudraten nicht nur sehr große Geschwindigkeiten, sondern arbeiten dazu auch noch sehr platzsparend.

Backup-Master ist mit seiner Menü- und Dialog-Führung leicht zu bedienen. Er gibt zu jedem geladenen Programm die Ladeadresse, Länge und Startadresse als hexadezimale Werte aus. Der Dateityp ist durch das gleiche Symbol dargestellt wie beim CAT-Befehl im Kassettenbetrieb.

Das Hauptmenü bietet Ihnen folgende Punkte zur Auswahl:
**Catalog:** Entspricht dem Basic-Befehl CAT und wirkt wahlweise auf Kassette oder Diskette.
**Baudrate einstellen:** Die Übertragungsgeschwindigkeit für Kassettenspeicherungen läßt sich im Bereich von 1000 bis 3500 Baud in 500-Baud-Schritten wählen.
**Header ausgeben:** Zeigt Ladeadresse, Länge, Startadresse und Dateityp eines Programms von Kassette oder Diskette an.
**Parameter ändern:** Im Bildschirm-Dialog ändern Sie in diesem Untermenü wichtige Voreinstellungen des Programms.

Wollen Sie den kompletten Inhalt einer Diskette oder Kassette für Backup-Zwecke kopieren, drücken Sie bei der Frage »Selektives oder totales Kopieren« die Taste <T>. Ihr CPC kopiert dann eine Datei nach der anderen, so wie er sie vorfindet, bis Sie mit <ESC> unterbrechen. Geht es Ihnen jedoch nur um einzelne Dateien, wählen Sie <S>. Bei Kassettenbetrieb gibt der Computer nacheinander die Namen aller gefundenen Dateien aus und erwartet die Freigabe des Kopiervorgangs. Ist die Diskette als Quelle angegeben, ruft der Backup-Master deren Directory auf den Bildschirm. Dort bewegen Sie den Cursor auf die gewünschten Dateinamen und markieren sie mit <COPY>.

Bei Speicherung auf Kassette sind Sie nicht an das Standardformat mit seinen 2-KByte-Blöcken gebunden. Wahlweise speichert Backup-Master beliebig große Dateien in jeweils nur zwei Blöcken. Das erhöht sowohl die Ladegeschwindigkeit als auch die Speicherkapazität der Kassetten.

Die Extension des Disketten-Dateinamens ist beim Transfer nach Wunsch zu übernehmen oder abzuschneiden.

Auch das Bezugslaufwerk und die Nummer des User-Bereichs lassen sich wechseln.

Am einfachsten ist, die Dateinamen zu übernehmen. Wollen Sie aber bereits beim Kopieren neue Namen vergeben, ändern Sie die Voreinstellung des letzten Untermenü-Punkts. Wichtig ist diese Funktion bei Kassettenprogrammen, deren Namen länger als acht Buchstaben sind und die auf Diskette übertragen werden sollen.
**Datei löschen:** Entspricht in seiner Wirkung dem RSX-Befehl ERA und wirkt somit nur auf Diskette.
**Datei umbenennen:** Entspricht dem Basic-Befehl REN.
**Ende:** Abbruch des Programms mit einem Reset.

Auch besonders lange Programme oder solche, die das Floppy-RAM im Bereich von A64D bis AC00 hex belegen, sind auf Diskette zu transferieren (lesen Sie zu dieser Problematik im 4. Schneider-Sonderheft von Happy-Computer (SH 7/86) den Beitrag »Schwertransport«). Das funktioniert, weil der Maschinencode des Backup-Master in einem Speicherbereich liegt, den normalerweise keine anderen Programme nutzen: im Bildschirmspeicher. Er reserviert sich dort einfach die untersten sechs Zeilen. Gespeicherte Bildschirminhalte und andere Dateien aus diesem Bereich lädt Backup-Master deshalb an eine andere Adresse.

Geben Sie bitte zuerst Listing 1 ein. Es enthält den Basic-Lader für den nötigen Maschinencode, den es nach dem Start automatisch unter dem Namen »BACKUP.BIN« speichert. Wenn Sie sich einen Gefallen tun wollen, erledigen Sie die Eingabe mit unserem speziellen Checksummer »CPC«. Er macht die Eingabe nicht nur absolut sicher, er verkürzt auch Ihre Tipparbeit drastisch, weil Sie nur noch gut die Hälfte des gesamten Listingumfangs zu verarbeiten haben. Lesen Sie bitte die entsprechenden Hinweise auf Seite 84.

Listing 2 ist die kleine Routine zum Laden und Aktivieren des Maschinencodes. Arbeiten Sie mit Kassetten, speichern Sie die Binärdatei »BACKUP.BIN« hinter der Laderoutine.

Aber vergessen Sie bitte bei all der Kopiererei nicht, daß Kopien immer nur für den eigenen Bedarf fair und erlaubt sind.

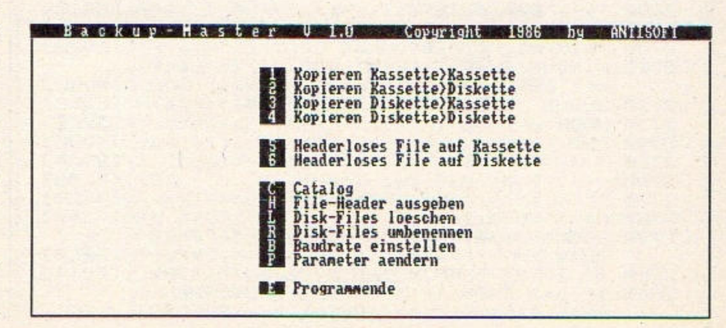
(Gerd Weinand/ja)

| Steckbrief | |
|---|---|
| Programm: | Backup-Master |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora/CPC |
| Datenträger: | Kassette/Diskette |

```
Backup-Master V 1.0      Copyright 1986 by ANTISOFT

   Selektiv oder total kopieren (S/T)? S
   Speichern in 2 Bloecken (J/N)? J
   Extension mit abspeichern (J/N)? J
   Defaultlaufwerk (A-D)? A
   Userbereich (0-F)? 0
   Filename aendern (J/N)? █
```

Diverse Parameter erlauben Anpassungen




Das Hauptmenü zeigt die Fähigkeiten






```
100 '*********************************** [31D4]
101 '* BACKUP.DAT - DATA-Lader von 'CPC' * [E2FA]
102 '*********************************** [A3D8]
103 ' [DEB6]
104 DATA 8000,CD,4E,BB,3E,02,CD,0E,BC,6204 [2ADC]
105 DATA 8008,97,47,4F,C5,CD,32,BC,C1,58E9 [C6D2]
106 DATA 8010,CD,38,BC,3E,01,01,1A,1A,7CC2 [B194]
107 DATA 8018,CD,32,BC,11,0A,00,21,86,7C04 [DA44]
108 DATA 8020,01,CD,C0,BB,21,08,81,11,204B [882C]
109 DATA 8028,09,06,06,0D,CD,80,80,11,0069 [1B06]
110 DATA 8030,B6,00,21,D2,00,CD,C0,BB,500F [B05A]
111 DATA 8038,21,15,81,11,02,02,06,11,04F5 [5396]
112 DATA 8040,3E,55,32,38,B3,CD,80,80,08AC [EC60]
113 DATA 8048,11,00,00,21,78,00,CD,C0,080A [BCEC]
114 DATA 8050,BB,21,26,81,11,06,0A,06,5992 [E0F2]
115 DATA 8058,08,3E,FD,32,38,B3,CD,80,1516 [137A]
116 DATA 8060,80,11,84,00,21,14,00,CD,5555 [66CA]
117 DATA 8068,C0,BB,21,2E,81,11,01,01,4C4F [7D48]
118 DATA 8070,06,2E,3E,FF,32,38,B3,CD,006B [91A8]
119 DATA 8078,80,80,CD,06,BB,C3,B1,81,7EF7 [D9C8]
120 DATA 8080,F3,E5,D5,CD,C6,BB,ED,53,53E5 [5318]
121 DATA 8088,A8,AC,22,AA,AC,D1,26,00,7788 [F5A8]
122 DATA 8090,6A,22,A4,AC,6B,22,A6,AC,21F0 [ECA0]
123 DATA 8098,E1,C5,E5,7E,CD,A5,BB,06,5F0C [591C]
124 DATA 80A0,08,C5,CD,06,B9,7E,E5,2A,2950 [9AAE]
125 DATA 80A8,A6,AC,45,C5,06,08,07,30,7CDE [73AC]
126 DATA 80B0,10,C5,F5,21,00,00,ED,5B,2471 [793C]
127 DATA 80B8,A4,AC,CD,F9,BB,F1,C1,18,68B6 [CE3E]
128 DATA 80C0,0B,ED,5B,2C,B3,2A,A4,AC,33B4 [C700]
129 DATA 80C8,19,22,2C,B3,10,E0,2A,2E,09CA [F294]
130 DATA 80D0,B3,2B,2B,22,2E,B3,ED,5B,563D [6CCA]
131 DATA 80D8,A8,AC,ED,53,2C,B3,C1,10,65AE [DEFA]
132 DATA 80E0,CA,E1,23,C1,10,BB,ED,5B,56DD [F908]
133 DATA 80E8,A4,AC,21,08,00,CD,BE,BD,7F55 [7002]
134 DATA 80F0,ED,5B,A8,AC,19,22,A8,AC,7E3C [552C]
135 DATA 80F8,22,2C,B3,2A,AA,AC,22,2E,094A [BCCA]
136 DATA 8100,B3,E1,23,C1,10,93,FB,C9,6A43 [9A78]
137 DATA 8108,42,61,63,6B,75,70,2D,4D,31EF [6260]
138 DATA 8110,61,73,74,65,72,63,6F,70,2622 [33E8]
139 DATA 8118,79,72,69,67,68,74,20,31,29B1 [090E]
140 DATA 8120,39,38,36,20,62,79,41,4E,1478 [30E4]
141 DATA 8128,54,49,53,4F,46,54,28,20,35C0 [FD02]
142 DATA 8130,47,65,72,64,20,57,65,69,323F [28F0]
143 DATA 8138,6E,61,6E,64,20,2F,20,48,2574 [0626]
144 DATA 8140,65,72,72,65,6E,73,74,72,2436 [C9F8]
145 DATA 8148,2E,20,31,34,20,2F,20,35,1BA9 [9014]
146 DATA 8150,35,39,30,20,43,6F,63,68,13CA [4814]
147 DATA 8158,65,6D,20,29,04,32,18,20,2FA8 [2D1E]
148 DATA 8160,20,20,20,42,20,61,20,63,1887 [AD94]
149 DATA 8168,20,6B,20,75,20,70,20,2D,093D [9008]
150 DATA 8170,20,4D,20,61,20,73,20,74,01A8 [19C4]
151 DATA 8178,20,65,20,72,20,20,20,56,0BF6 [D2D8]
152 DATA 8180,20,20,31,2E,30,20,20,20,1DA0 [A5BA]
153 DATA 8188,20,20,20,43,6F,70,79,72,1A08 [80F6]
154 DATA 8190,69,67,68,74,20,20,20,31,26F1 [64E0]
155 DATA 8198,39,38,36,20,20,20,62,79,177D [9EF8]
156 DATA 81A0,20,20,20,41,4E,54,49,53,1BF1 [BA10]
157 DATA 81A8,4F,46,54,20,20,20,20,18,3F58 [8F12]
158 DATA 81B0,00,3E,FF,32,E8,B4,32,22,1696 [D874]
159 DATA 81B8,B6,32,23,B6,21,40,00,22,58AA [D838]
160 DATA 81C0,20,B6,21,5C,81,7E,23,B7,3961 [D748]
161 DATA 81C8,28,05,CD,5A,BB,18,F6,3E,0D2A [C2EC]
162 DATA 81D0,06,01,00,BC,ED,79,3E,13,0E63 [4B76]
163 DATA 81D8,01,00,BD,ED,79,21,34,82,1A56 [476C]
164 DATA 81E0,11,F0,C5,01,10,02,ED,B0,2DD2 [EC74]
165 DATA 81E8,11,F0,CD,01,10,02,ED,B0,2CD2 [6DA2]
166 DATA 81F0,11,F0,D5,01,10,02,ED,B0,2FD2 [9380]
167 DATA 81F8,11,F0,DD,01,10,02,ED,B0,2ED2 [DEAE]
168 DATA 8200,11,F0,E5,01,10,02,ED,B0,29D2 [2342]
169 DATA 8208,11,F0,ED,01,10,02,ED,B0,28D2 [2670]
170 DATA 8210,11,F0,F5,01,10,02,ED,B0,2BD2 [164A]
171 DATA 8218,11,F0,FD,01,10,02,ED,B0,2AD2 [0178]
172 DATA 8220,21,02,00,11,12,50,CD,66,113C [1ECC]
173 DATA 8228,BB,CD,06,B9,C3,F0,C5,00,61C2 [F3C2]
174 DATA 8230,00,00,00,00,21,F0,CD,CD,039F [2E30]
175 DATA 8238,49,C6,CD,7D,BC,CD,06,BB,0D13 [7E1C]
176 DATA 8240,FE,45,CA,CD,F7,CD,F0,ED,7E51 [1B6C]
177 DATA 8248,38,F3,F5,CD,6C,BB,F1,FE,3220 [720E]
178 DATA 8250,4C,CC,E3,EE,FE,52,CC,37,0097 [550C]
179 DATA 8258,EF,FE,31,CC,3E,DE,FE,32,41A6 [0B56]
180 DATA 8260,CC,81,DE,FE,33,CC,33,DF,5071 [50F6]
181 DATA 8268,FE,34,CC,F0,E5,FE,35,CC,60F6 [AF36]
182 DATA 8270,B1,DF,FE,36,CC,88,FE,FE,76A2 [8D5E]
183 DATA 8278,43,CC,DD,E6,FE,42,CC,99,00B9 [4F1E]
184 DATA 8280,EF,FE,50,CC,09,EE,FE,48,4C84 [2646]
185 DATA 8288,CC,4C,F7,18,A7,7E,23,B7,6E51 [59E8]
186 DATA 8290,C8,CD,5A,BB,18,F7,3E,0A,54DA [4A22]
187 DATA 8298,CD,5A,BB,3E,0D,C3,5A,BB,67EB [CE64]
188 DATA 82A0,06,0B,21,00,01,23,97,BC,04F6 [8F30]
189 DATA 82A8,C2,61,C6,10,F5,C9,18,C0,64FC [3CCE]
190 DATA 82B0,07,1B,C0,07,0F,C0,07,12,1E54 [A844]
191 DATA 82B8,C0,07,21,BC,B8,06,10,CD,6815 [4272]
192 DATA 82C0,82,C7,ED,5B,20,B6,CD,8C,6A1E [BA18]
193 DATA 82C8,BC,3A,2F,B6,FE,10,38,06,59C6 [B3EA]
194 DATA 82D0,3A,23,B6,B7,20,1D,CD,B5,082B [3BBA]
195 DATA 82D8,E7,3E,02,32,CC,B8,3A,2B,7BBF [411E]
196 DATA 82E0,B6,2A,2C,B6,ED,5B,2E,B6,598E [6E2E]
197 DATA 82E8,ED,4B,30,B6,CD,98,BC,D0,6C80 [A134]
198 DATA 82F0,C3,8F,BC,CD,B5,E7,3E,01,5F59 [7D3A]
199 DATA 82F8,32,5C,B8,97,32,5D,B8,21,11C5 [CE9A]
200 DATA 8300,00,08,22,5F,B8,2A,2C,B6,0636 [2E26]
201 DATA 8308,22,61,B8,3D,32,63,B8,3A,1CC6 [B362]
202 DATA 8310,2B,B6,32,5E,B8,2A,2E,B6,3E22 [0C92]
203 DATA 8318,22,64,B8,2A,30,B6,22,66,1EDA [2060]
204 DATA 8320,B8,11,4C,B8,06,8A,CD,06,59C4 [498E]
205 DATA 8328,B9,CD,95,26,CD,09,B9,21,781F [1BBC]
206 DATA 8330,4C,B8,11,40,00,3E,2C,CD,0E4D [FC88]
207 DATA 8338,9E,BC,D0,2A,2C,B6,E5,11,7AC3 [34EC]
208 DATA 8340,00,08,D5,3E,16,CD,9E,BC,1944 [3596]
209 DATA 8348,D1,E1,D0,19,22,61,B8,E5,4A51 [0876]
210 DATA 8350,2A,64,B8,ED,52,22,5F,B8,17CE [78B2]
211 DATA 8358,E5,97,32,63,B8,3D,32,5D,523D [D06E]
212 DATA 8360,B8,21,5C,B8,34,11,4C,B8,5584 [F556]
213 DATA 8368,06,8A,CD,06,B9,CD,95,26,3FB0 [27C8]
214 DATA 8370,CD,09,B9,21,4C,B8,11,40,7112 [E03E]
215 DATA 8378,00,3E,2C,CD,9E,BC,D1,E1,0193 [40CE]
216 DATA 8380,D0,3E,16,CD,9E,BC,C3,92,6E84 [51EE]
217 DATA 8388,BC,3E,01,32,CC,B8,ED,5B,5781 [27E6]
218 DATA 8390,20,B6,CD,77,BC,D0,22,32,2586 [E276]
219 DATA 8398,B6,32,2B,B6,11,15,00,19,5945 [2F20]
220 DATA 83A0,7E,32,2C,B6,23,7E,FE,C0,3CBC [79FC]
221 DATA 83A8,38,02,3E,40,32,2D,B6,23,1F2B [6456]
222 DATA 83B0,23,7E,32,2E,B6,23,7E,32,0F52 [4A62]
223 DATA 83B8,2F,B6,23,7E,32,30,B6,23,399F [3E86]
224 DATA 83C0,7E,32,31,B6,97,C9,3A,39,3911 [7066]
225 DATA 83C8,B6,B7,C8,E5,C5,F5,21,FD,64D3 [7718]
226 DATA 83D0,D6,CD,49,C6,CD,81,BB,CD,58D7 [4452]
227 DATA 83D8,C2,EE,47,F1,32,39,B6,78,5D10 [86B6]
228 DATA 83E0,B7,28,1A,18,02,C5,C5,CD,5103 [EE84]
229 DATA 83E8,52,C6,CD,D9,EF,CD,52,C6,089E [B24E]
230 DATA 83F0,CD,52,C6,28,08,C1,C1,E5,6A63 [ABAE]
231 DATA 83F8,CD,8D,EF,E1,C9,C1,E1,37,5289 [CC22]
232 DATA 8400,C9,CD,7A,BC,3E,28,32,86,52F2 [17BA]
233 DATA 8408,B2,21,85,B2,35,3A,2B,B6,4B60 [135C]
234 DATA 8410,C6,24,CD,5A,BB,21,2C,B6,73B2 [38B4]
235 DATA 8418,CD,E8,C7,21,2E,B6,CD,E8,44AA [3426]
236 DATA 8420,C7,21,30,B6,CD,E8,C7,3E,62D8 [9BC2]
237 DATA 8428,0A,C3,5A,BB,3E,20,CD,5A,3580 [2AC0]
238 DATA 8430,BB,01,02,04,D7,19,31,EB,5B95 [5064]
239 DATA 8438,C3,49,C6,7E,23,CD,5A,BB,6EC3 [FD14]
240 DATA 8440,10,F9,C9,00,0C,1F,1C,02,2F46 [B754]
241 DATA 8448,18,20,31,20,18,20,4B,6F,0099 [F5EA]
242 DATA 8450,70,69,65,72,65,6E,20,4B,2B5B [C744]
243 DATA 8458,61,73,73,65,74,74,65,3E,26F4 [3332]
244 DATA 8460,4B,61,73,73,65,74,74,65,36E5 [6424]
245 DATA 8468,1F,1C,03,18,20,32,20,18,08F0 [1404]
246 DATA 8470,20,4B,6F,70,69,65,72,65,0A7D [8554]
247 DATA 8478,6E,20,4B,61,73,73,65,74,329A [9D42]
248 DATA 8480,74,65,3E,44,69,73,6B,65,22F7 [E256]
249 DATA 8488,74,74,65,1F,1C,04,18,20,2AB0 [7542]
250 DATA 8490,33,20,18,20,4B,6F,70,69,13ED [EF34]
251 DATA 8498,65,72,65,6E,20,44,69,73,24F1 [9A2A]
252 DATA 84A0,6B,65,74,74,65,3E,4B,61,2627 [8D4C]
253 DATA 84A8,73,73,65,74,74,65,1F,1C,2CB6 [EC7A]
254 DATA 84B0,05,18,20,34,20,18,20,4B,02AB [C3FC]
255 DATA 84B8,6F,70,69,65,72,65,6E,20,2208 [A344]
256 DATA 84C0,44,69,73,6B,65,74,74,65,32E5 [374C]
257 DATA 84C8,3E,44,69,73,6B,65,74,74,0640 [2A4E]
258 DATA 84D0,65,1F,1C,07,18,20,35,20,36BA [2A46]
259 DATA 84D8,18,20,48,65,61,64,65,72,0970 [3B0A]
260 DATA 84E0,6C,6F,73,65,73,20,46,69,260D [ED54]
261 DATA 84E8,6C,65,20,61,75,66,20,4B,2F6B [A96A]
262 DATA 84F0,61,73,73,65,74,74,65,1F,26D5 [B142]
263 DATA 84F8,1C,08,18,20,36,20,18,20,0C20 [71FA]
264 DATA 8500,48,65,61,64,65,72,6C,6F,3577 [192C]
265 DATA 8508,73,65,73,20,46,69,6C,65,2F89 [CE3E]
266 DATA 8510,20,61,75,66,20,44,69,73,0031 [21B4]
267 DATA 8518,6B,65,74,74,65,1F,1C,0A,2666 [D15E]
268 DATA 8520,18,20,43,20,18,20,43,61,0EC7 [89E2]
269 DATA 8528,74,61,6C,6F,67,1F,1C,0B,2A47 [349A]
270 DATA 8530,18,20,48,20,18,20,46,69,0FA5 [3BF0]
271 DATA 8538,6C,65,2D,48,65,61,64,65,2C61 [DA42]
272 DATA 8540,72,20,61,75,73,67,65,62,38DC [4318]
273 DATA 8548,65,6E,1F,1C,0C,18,20,4C,2B2C [F994]
274 DATA 8550,20,18,20,44,69,73,6B,2D,143F [5F1E]
275 DATA 8558,46,69,6C,65,73,20,6C,6F,313F [7A72]
276 DATA 8560,65,73,63,68,65,6E,1F,1C,2612 [2540]
277 DATA 8568,0D,18,20,52,20,18,20,44,00C4 [5DE6]
278 DATA 8570,69,73,6B,2D,46,69,6C,65,24D9 [9B84]
279 DATA 8578,73,20,75,6D,62,65,6E,65,3BCD [DF96]
280 DATA 8580,6E,6E,65,6E,1F,1C,0E,18,264C [B4B2]
281 DATA 8588,20,42,20,18,20,42,61,75,05BF [D5F2]
282 DATA 8590,64,72,61,74,65,20,65,69,26EB [8E26]
283 DATA 8598,6E,73,74,65,6C,6C,65,6E,2164 [8282]
284 DATA 85A0,1F,1C,0F,18,20,50,20,18,08F8 [6048]
285 DATA 85A8,20,50,61,72,61,6D,65,74,0D02 [831E]
286 DATA 85B0,65,72,20,61,65,6E,64,65,2E2D [D954]
287 DATA 85B8,72,6E,1F,1C,11,18,20,45,204D [735A]
288 DATA 85C0,20,18,20,50,72,6F,67,72,1590 [5508]
289 DATA 85C8,61,6D,6D,65,6E,64,65,07,221D [6A8E]
290 DATA 85D0,00,41,20,6F,64,65,72,20,10C0 [3EF2]
291 DATA 85D8,42,3F,07,20,00,31,2E,20,2C98 [F52C]
292 DATA 85E0,31,30,30,30,20,42,61,75,113F [22D0]
293 DATA 85E8,64,0A,0A,0D,32,2E,20,31,3049 [9C40]
294 DATA 85F0,35,30,30,20,42,61,75,64,119A [B9EE]
295 DATA 85F8,0A,0A,0D,33,2E,20,32,30,04B4 [9250]
296 DATA 8600,30,30,20,42,61,75,64,0A,163E [63E0]
297 DATA 8608,0A,0D,34,2E,20,32,35,30,03B2 [ED14]
298 DATA 8610,30,20,42,61,75,64,0A,0A,1C76 [9F06]
299 DATA 8618,0D,35,2E,20,33,30,30,30,0D08 [3902]
300 DATA 8620,20,42,61,75,64,0A,0A,0D,08E1 [0704]
301 DATA 8628,36,2E,20,33,35,30,30,20,1698 [09C4]
302 DATA 8630,42,61,75,64,0A,0A,0D,42,3180 [71EE]
303 DATA 8638,69,74,74,65,20,77,61,65,212B [00FC]
304 DATA 8640,68,6C,65,6E,20,53,69,65,25BB [3456]
305 DATA 8648,20,28,31,2D,36,29,3A,20,1FB0 [3E22]
306 DATA 8650,07,00,00,00,0A,0D,42,69,0309 [02BE]
```

**Listing 1. Gute $4\frac{1}{2}$ KByte machen aus Ihrem CPC eine »Kopiermaschine«**

```
307 DATA 8658,74,74,65,20,51,75,65,6C,2A5A  [FC3A]
308 DATA 8660,6C,6B,61,73,73,65,74,74,2540  [A21A]
309 DATA 8668,65,20,65,69,6E,6C,65,67,32DD  [9B7C]
310 DATA 8670,65,6E,2C,20,50,4C,41,59,2DEB  [6F70]
311 DATA 8678,20,75,6E,64,20,65,69,6E,06E8  [E74A]
312 DATA 8680,65,20,54,61,73,74,65,20,34B2  [89D8]
313 DATA 8688,64,72,75,65,63,6B,65,6E,2460  [0830]
314 DATA 8690,2E,0A,0D,07,00,43,61,74,15EA  [C53A]
315 DATA 8698,61,6C,6F,67,20,76,6F,6E,2078  [117C]
316 DATA 86A0,20,44,69,73,6B,65,74,74,0940  [CF10]
317 DATA 86A8,65,20,6F,64,65,72,20,4B,33CB  [7364]
318 DATA 86B0,61,73,73,65,74,74,65,20,26EA  [6932]
319 DATA 86B8,28,44,2F,4B,29,3F,20,07,05A3  [4762]
320 DATA 86C0,00,53,65,6C,65,6B,74,69,1CA5  [985E]
321 DATA 86C8,76,20,6F,64,65,72,20,74,3A74  [2C34]
322 DATA 86D0,6F,74,61,6C,20,6B,6F,70,2062  [A35A]
323 DATA 86D8,69,65,72,65,6E,20,28,53,2623  [5B24]
324 DATA 86E0,2F,54,29,3F,20,00,0A,0A,054E  [AF52]
325 DATA 86E8,0D,53,70,65,69,63,68,65,1861  [B734]
326 DATA 86F0,72,6E,20,69,6E,20,32,20,23A4  [A730]
327 DATA 86F8,42,6C,6F,65,63,6B,65,6E,33A0  [62A4]
328 DATA 8700,20,28,4A,2F,4E,29,3F,20,133A  [AD44]
329 DATA 8708,00,0A,0A,0D,45,78,74,65,0055  [2E12]
330 DATA 8710,6E,73,69,6F,6E,20,6D,69,2353  [945E]
331 DATA 8718,74,20,61,62,73,70,65,69,3AFB  [E830]
332 DATA 8720,63,68,65,72,6E,20,28,4A,23EA  [EF42]
333 DATA 8728,2F,4E,29,3F,20,00,0A,0A,03CE  [2B78]
334 DATA 8730,0D,44,65,66,61,75,6C,74,1F30  [7A2C]
335 DATA 8738,6C,61,75,66,77,65,72,6B,2423  [2530]
336 DATA 8740,20,28,41,2D,44,29,3F,20,122A  [F800]
337 DATA 8748,00,0A,0A,0D,55,73,65,72,00CC  [0A44]
338 DATA 8750,62,65,72,65,69,63,68,20,2264  [A2EA]
339 DATA 8758,28,30,2D,46,29,3F,20,00,1834  [2808]
340 DATA 8760,0A,0A,0D,46,69,6C,65,6E,001C  [6168]
341 DATA 8768,61,6D,65,20,61,65,6E,64,2744  [DF22]
342 DATA 8770,65,72,6E,20,28,4A,2F,4E,21B8  [C166]
343 DATA 8778,29,3F,20,00,0A,0A,0D,4E,1F6C  [3386]
344 DATA 8780,6F,63,68,20,65,69,6E,20,2230  [4616]
345 DATA 8788,46,69,6C,65,20,6B,6F,70,3292  [5056]
346 DATA 8790,69,65,72,65,6E,20,28,4A,263A  [9642]
347 DATA 8798,2F,4E,29,3F,20,07,00,0A,03C6  [005E]
348 DATA 87A0,0A,0D,42,69,74,74,65,20,0A0A  [3748]
349 DATA 87A8,51,75,65,6C,6C,64,69,73,3DF1  [FD9C]
350 DATA 87B0,6B,65,74,74,65,20,69,6E,2614  [3138]
351 DATA 87B8,20,4C,61,75,66,77,65,72,0A24  [9D32]
352 DATA 87C0,6B,20,00,0A,0A,0D,42,69,3DA9  [0F70]
353 DATA 87C8,74,74,65,20,5A,69,65,6C,2A72  [2166]
354 DATA 87D0,64,69,73,6B,65,74,74,65,22E5  [A054]
355 DATA 87D8,20,69,6E,20,4C,61,75,66,06E8  [A66A]
356 DATA 87E0,77,65,72,6B,20,00,0A,0D,2B29  [AA50]
357 DATA 87E8,14,4C,6F,61,64,69,6E,67,10CF  [A5B6]
358 DATA 87F0,20,00,20,65,69,6E,6C,65,101D  [3A3C]
359 DATA 87F8,67,65,6E,20,75,6E,64,20,27F8  [5186]
360 DATA 8800,65,69,6E,65,20,54,61,73,23B1  [5706]
361 DATA 8808,74,65,20,64,72,75,65,63,23ED  [B922]
362 DATA 8810,6B,65,6E,2E,0A,0D,07,00,238A  [E262]
363 DATA 8818,0A,0D,14,42,69,74,74,65,02F5  [2430]
364 DATA 8820,20,5A,69,65,6C,6B,61,73,0F8D  [8376]
365 DATA 8828,73,65,74,74,65,20,65,69,2A0B  [1D2A]
366 DATA 8830,6E,6C,65,67,65,6E,2C,20,2438  [9A64]
367 DATA 8838,52,45,43,2B,50,4C,41,59,31FB  [914E]
368 DATA 8840,20,75,6E,64,20,65,69,6E,06E8  [6450]
369 DATA 8848,65,20,54,61,73,74,65,20,34B2  [4CFC]
370 DATA 8850,64,72,75,65,63,6B,65,6E,2460  [C524]
371 DATA 8858,2E,0A,0D,07,00,00,00,00,1450  [94D6]
372 DATA 8860,00,00,00,00,46,69,6C,65,0329  [3BC6]
373 DATA 8868,2D,48,65,61,64,65,72,00,0C60  [0E18]
374 DATA 8870,0A,0D,14,53,61,76,69,6E,039C  [A454]
375 DATA 8878,67,20,20,00,42,69,74,74,3CA8  [971E]
376 DATA 8880,65,20,46,69,6C,65,6E,61,3699  [854A]
377 DATA 8888,6D,65,20,65,69,6E,67,65,2FCB  [22B0]
378 DATA 8890,62,65,6E,3A,20,07,00,42,277E  [2A2E]
379 DATA 8898,69,74,74,65,20,6E,65,75,2157  [163A]
380 DATA 88A0,65,6E,00,0A,0A,0D,42,69,2929  [6752]
381 DATA 88A8,74,74,65,20,61,6C,74,65,2B95  [E24C]
382 DATA 88B0,6E,00,97,DF,6A,C6,97,DF,29A9  [D412]
383 DATA 88B8,6D,C6,21,F0,D5,CD,49,C6,09E8  [DF0C]
384 DATA 88C0,CD,06,BB,06,00,CD,45,C7,7379  [30C6]
385 DATA 88C8,CD,2D,FF,21,3E,DE,DA,D4,73D8  [F982]
386 DATA 88D0,DF,38,F0,2A,2C,B6,CD,83,7F81  [AF1C]
387 DATA 88D8,BC,D0,CD,BD,C7,21,B4,D7,7F73  [0A5E]
388 DATA 88E0,CD,49,C6,CD,06,BB,CD,76,63E0  [123C]
389 DATA 88E8,C6,3A,22,B6,B7,28,C3,21,661F  [16BE]
390 DATA 88F0,3E,DE,C3,D4,DF,97,DF,6A,38D0  [5D62]
391 DATA 88F8,C6,97,DF,73,C6,21,5F,D7,5CCD  [FD3E]
392 DATA 8900,CD,49,C6,3A,D8,A6,C6,41,6A35  [5ECC]
393 DATA 8908,CD,5A,BB,21,8E,D7,CD,49,638F  [372A]
394 DATA 8910,C6,CD,06,BB,21,F0,D5,CD,589F  [3D0E]
395 DATA 8918,49,C6,CD,06,BB,06,00,CD,09CD  [E0EA]
396 DATA 8920,45,C7,CD,2D,FF,21,A9,DE,0EC0  [0E2A]
397 DATA 8928,DA,D4,DF,38,F0,CD,18,E7,4403  [6B06]
398 DATA 8930,D2,1B,DF,CD,83,BC,D0,CD,7F75  [A158]
399 DATA 8938,BD,C7,2A,32,B6,7E,B7,20,6C26  [40DE]
400 DATA 8940,06,CD,A1,C7,C3,03,DF,11,2FAB  [96D0]
401 DATA 8948,A4,AC,01,08,00,7E,FE,2E,798A  [F2C2]
402 DATA 8950,28,0E,FE,00,28,15,ED,A0,082E  [898A]
403 DATA 8958,EA,D9,DE,7E,FE,2E,20,0B,5863  [2B1E]
404 DATA 8960,3E,08,91,01,04,00,81,ED,0EFF  [CB6E]
405 DATA 8968,B0,18,03,3E,08,91,47,97,5F9D  [3872]
406 DATA 8970,12,21,A4,AC,CD,82,C7,E5,1A0B  [DAC0]
407 DATA 8978,21,FD,DD,CD,49,C6,E1,E5,38C7  [9830]
408 DATA 8980,CD,49,C6,E1,ED,5B,20,B6,64E2  [40F2]
409 DATA 8988,CD,8C,BC,CD,92,C6,D0,CD,5835  [1F4C]
410 DATA 8990,5C,C6,3A,22,B6,B7,F5,CC,1CAA  [170C]
411 DATA 8998,52,C6,CC,52,C6,F1,CA,A9,00E9  [70EE]
412 DATA 89A0,DE,21,A9,DE,C3,D4,DF,97,7BE1  [264A]
413 DATA 89A8,DF,6D,C6,97,DF,70,C6,21,63E5  [E70A]
414 DATA 89B0,3B,D7,CD,49,C6,3A,D8,A6,32BE  [5C24]
415 DATA 89B8,C6,41,CD,5A,BB,21,8E,D7,6BD7  [6030]
416 DATA 89C0,CD,49,C6,CD,06,BB,CD,F0,6366  [CD2A]
417 DATA 89C8,F5,C5,E5,21,B4,D7,CD,49,525F  [B00A]
418 DATA 89D0,C6,CD,06,BB,E1,C1,C5,E5,5E53  [5A20]
419 DATA 89D8,21,82,D7,CD,49,C6,E1,E5,2647  [4BD2]
420 DATA 89E0,06,0C,C5,CD,F7,C7,CD,52,111C  [4AEA]
421 DATA 89E8,C6,C1,E1,E5,CD,45,C7,2A,47E8  [7F1A]
422 DATA 89F0,2C,B6,CD,83,BC,E1,C1,D0,2D26  [150A]
423 DATA 89F8,C5,E5,CD,BD,C7,E1,E5,06,4DC0  [3F50]
424 DATA 8A00,0C,3A,24,B6,B7,20,10,01,0279  [0C1A]
425 DATA 8A08,09,00,09,3E,20,ED,A9,20,0586  [7F4C]
426 DATA 8A10,03,EA,99,DF,ED,42,41,CD,23FF  [101A]
427 DATA 8A18,7B,C6,E1,C1,D0,11,0C,00,1AEC  [A5C8]
428 DATA 8A20,19,10,B3,97,C9,97,DF,6D,1257  [4C9C]
429 DATA 8A28,C6,21,F0,D5,CD,49,C6,CD,7E1D  [A830]
430 DATA 8A30,06,BB,CD,F0,FD,21,B4,D7,3DB3  [1D14]
431 DATA 8A38,CD,49,C6,CD,06,BB,CD,E7,6371  [A428]
432 DATA 8A40,EF,D0,21,B1,DF,C3,D4,DF,4833  [F70A]
433 DATA 8A48,E5,21,18,D7,CD,49,C6,E1,7291  [A9B8]
434 DATA 8A50,E5,CD,06,BB,2A,85,B2,CD,485D  [F614]
435 DATA 8A58,5A,BB,22,85,B2,E1,FE,4E,0876  [80EC]
436 DATA 8A60,C8,FE,4A,C4,B0,E7,20,E8,5834  [DFEA]
437 DATA 8A68,CD,52,C6,CD,52,C6,37,E9,671F  [6A0A]
438 DATA 8A70,00,00,00,00,32,34,B6,97,00BB  [13FA]
439 DATA 8A78,DF,70,C6,97,DF,73,C6,21,64A9  [DFF8]
440 DATA 8A80,44,D7,CD,34,F7,48,CD,52,0A70  [2ABE]
441 DATA 8A88,C6,CD,52,C6,21,68,D7,CD,57AB  [F814]
442 DATA 8A90,34,F7,78,B9,20,06,F5,97,2335  [6E60]
443 DATA 8A98,32,34,B6,F1,32,36,B6,79,0D8D  [9B98]
444 DATA 8AA0,32,35,B6,97,CD,52,C6,21,0D7D  [C9B4]
445 DATA 8AA8,3B,D7,CD,49,C6,3A,35,B6,3374  [A0EE]
446 DATA 8AB0,32,D8,A6,C6,41,CD,5A,BB,3693  [EDF6]
447 DATA 8AB8,21,8E,D7,CD,49,C6,CD,06,25FC  [203A]
448 DATA 8AC0,BB,CD,F0,F5,C5,E5,21,82,7AEC  [A45A]
449 DATA 8AC8,D7,CD,49,C6,E1,E5,06,0C,591C  [351C]
450 DATA 8AD0,C5,CD,F7,C7,C1,E1,E5,CD,47DB  [007C]
451 DATA 8AD8,45,C7,2A,2C,B6,CD,83,BC,13FE  [B43C]
452 DATA 8AE0,E1,C1,D0,C5,E5,3E,0A,CD,5199  [600E]
453 DATA 8AE8,5A,BB,CD,BD,C7,3A,36,B6,17BA  [4872]
454 DATA 8AF0,32,D8,A6,3A,34,B6,B7,20,3A56  [D4AE]
455 DATA 8AF8,17,21,5F,D7,CD,49,C6,3A,03AA  [8104]
456 DATA 8B00,D8,A6,C6,41,CD,5A,BB,21,5F07  [05EC]
457 DATA 8B08,8E,D7,CD,49,C6,CD,06,BB,6A43  [A738]
458 DATA 8B10,E1,06,0C,CD,82,C7,C5,E5,7A33  [37D8]
459 DATA 8B18,21,FC,DD,CD,49,C6,E1,C1,38A3  [762A]
460 DATA 8B20,E5,C5,CD,F7,C7,C1,E1,E5,510B  [6B10]
461 DATA 8B28,CD,8C,BC,CD,92,C6,E1,C1,585B  [484A]
462 DATA 8B30,D0,11,0C,00,19,10,01,C9,6D83  [FA2A]
463 DATA 8B38,3A,35,B6,32,D8,A6,3A,34,01B8  [E794]
464 DATA 8B40,B6,B7,C2,40,E6,E5,21,3B,6E5D  [1CDA]
465 DATA 8B48,D7,CD,49,C6,3A,D8,A6,C6,5EBA  [9A60]
466 DATA 8B50,41,CD,5A,BB,21,8E,D7,CD,1163  [96F0]
467 DATA 8B58,49,C6,CD,06,BB,E1,C3,40,0B5A  [70FA]
468 DATA 8B60,E6,21,31,D6,CD,49,C6,97,7657  [A4B4]
469 DATA 8B68,DF,70,C6,CD,06,BB,FE,44,64F4  [EA46]
470 DATA 8B70,CC,5A,BB,28,1C,FE,4B,C4,662A  [442E]
471 DATA 8B78,B0,E7,20,EF,CD,5A,BB,97,6DD1  [BF4C]
472 DATA 8B80,DF,6A,C6,CD,52,C6,21,F0,602A  [40F0]
473 DATA 8B88,D5,CD,49,C6,3E,01,32,CC,5DDC  [5A30]
474 DATA 8B90,B8,ED,5B,20,B6,CD,9B,BC,692E  [3152]
475 DATA 8B98,CD,06,BB,C9,CD,09,B9,2A,7BE4  [C65C]
476 DATA 8BA0,2C,B6,E5,ED,5B,2E,B6,19,2AE5  [813E]
477 DATA 8BA8,11,4E,A6,ED,52,E1,D8,2A,019E  [2C02]
478 DATA 8BB0,32,B6,CD,82,C7,E5,2A,2C,20D4  [D8F2]
479 DATA 8BB8,B6,22,77,FF,2A,2E,B6,22,5236  [78D4]
480 DATA 8BC0,7A,FF,22,A2,FF,2A,30,B6,0B26  [06FC]
481 DATA 8BC8,22,7C,FF,3A,D7,A6,32,62,1666  [5ED0]
482 DATA 8BD0,FF,11,00,00,06,42,21,62,7AD8  [4156]
483 DATA 8BD8,FF,7E,23,83,5F,30,01,14,6E7E  [92DC]
484 DATA 8BE0,10,F7,21,A5,FF,73,23,72,3DB0  [64B2]
485 DATA 8BE8,E1,E5,CD,8D,EF,11,00,40,5FCC  [2246]
486 DATA 8BF0,E1,CD,8C,BC,06,7F,21,62,586C  [3F12]
487 DATA 8BF8,FF,7E,23,C5,CD,95,BC,C1,6DB5  [128A]
488 DATA 8C00,30,30,10,F5,01,00,08,2A,1962  [5AEA]
489 DATA 8C08,20,B6,7E,23,C5,CD,95,BC,34FA  [8212]
490 DATA 8C10,C1,30,1F,0B,79,B0,20,F2,6E6A  [CB9A]
491 DATA 8C18,2A,2E,B6,7C,FE,08,38,0A,082A  [1BDC]
492 DATA 8C20,CD,80,BC,30,05,CD,95,BC,508A  [06EC]
493 DATA 8C28,38,F6,CD,8F,BC,CD,7A,BC,364C  [4180]
494 DATA 8C30,B7,C9,E1,C9,3E,07,C3,5A,7940  [56D6]
495 DATA 8C38,BB,E5,D5,C5,21,B2,FF,11,701F  [C700]
496 DATA 8C40,68,B8,01,0E,00,ED,B0,21,1835  [406A]
497 DATA 8C48,D2,FF,01,0B,00,ED,B0,C1,5445  [95CE]
498 DATA 8C50,D1,E1,C9,00,00,00,00,00,49E0  [7128]
499 DATA 8C80,00,00,00,00,FE,42,C8,FE,0796  [3278]
500 DATA 8C88,43,C8,FE,4C,C8,FE,48,C8,0D60  [9522]
501 DATA 8C90,FE,50,C8,FE,52,C8,FE,31,7D9D  [3D3E]
502 DATA 8C98,D8,FE,37,3F,C9,21,5D,D6,5030  [E2DA]
503 DATA 8CA0,CD,49,C6,E5,CD,81,BB,CD,6787  [8150]
504 DATA 8CA8,06,BB,FE,53,32,22,B6,28,376C  [74BC]
505 DATA 8CB0,0F,FE,54,F5,97,32,22,B6,3952  [E2AC]
506 DATA 8CB8,F1,28,05,CD,B0,E7,18,E7,783B  [B2F4]
507 DATA 8CC0,CD,5A,BB,E1,CD,49,C6,E5,6F55  [DF68]
508 DATA 8CC8,CD,06,BB,FE,4A,32,23,B6,7DE8  [E448]
509 DATA 8CD0,28,0F,FE,4E,F5,97,32,23,0953  [2DBA]
510 DATA 8CD8,B6,F1,28,05,CD,B0,E7,18,676E  [DFEE]
511 DATA 8CE0,E7,CD,5A,BB,E1,CD,49,C6,4458  [4F44]
512 DATA 8CE8,E5,CD,06,BB,FE,4A,32,24,4C28  [BF1A]
513 DATA 8CF0,B6,28,0F,FE,4E,F5,97,32,5FB8  [8B22]
514 DATA 8CF8,24,B6,F1,28,05,CD,B0,E7,21BB  [16F2]
515 DATA 8D00,18,E7,CD,5A,BB,E1,CD,49,2E4F  [DD3C]
516 DATA 8D08,C6,E5,CD,06,BB,D6,41,32,45B0  [2DD2]
```

```
517 DATA 8D10,D8,A6,B7,28,09,FE,04,38,5260  [F98C]
518 DATA 8D18,05,CD,B0,E7,18,EC,C6,41,2B0D  [49F4]
519 DATA 8D20,CD,5A,BB,E1,CD,49,C6,E5,6F55  [BC4E]
520 DATA 8D28,CD,06,BB,D6,30,FE,00,38,7F40  [34D0]
521 DATA 8D30,0E,FE,0A,38,0F,D6,07,FE,3990  [7BE0]
522 DATA 8D38,0A,38,04,FE,10,38,05,CD,04C7  [438E]
523 DATA 8D40,B0,E7,18,E4,32,D7,A6,C6,6FC6  [D5EA]
524 DATA 8D48,30,FE,39,38,02,C6,07,CD,20EB  [EEC0]
525 DATA 8D50,5A,BB,E1,CD,49,C6,CD,06,13FC  [0A34]
526 DATA 8D58,BB,FE,4A,20,09,32,39,B6,6904  [14AE]
527 DATA 8D60,CD,5A,BB,C3,84,BB,FE,4E,6C2E  [AE82]
528 DATA 8D68,C4,B0,E7,20,E9,CD,5A,BB,5493  [3514]
529 DATA 8D70,97,32,39,B6,C3,84,BB,CD,4EF3  [1702]
530 DATA 8D78,D9,EF,CD,8D,EF,32,08,EF,417F  [7672]
531 DATA 8D80,21,06,EF,11,00,40,01,05,0CF7  [D426]
532 DATA 8D88,00,ED,B0,3E,01,DD,21,00,2D9E  [2ABE]
533 DATA 8D90,40,DF,0B,EF,DF,70,C6,C3,1E27  [1F20]
534 DATA 8D98,0D,E7,02,40,00,A4,AC,2A,38E2  [31A6]
535 DATA 8DA0,C0,07,2D,C0,07,E5,E5,06,6A00  [EB96]
536 DATA 8DA8,0C,CD,60,BB,F5,CD,9C,BB,37EF  [6EA8]
537 DATA 8DB0,F1,E1,CD,5A,BB,E5,CD,9C,5B8A  [BD9C]
538 DATA 8DB8,BB,10,EE,E1,E1,C9,CD,60,4F86  [6650]
539 DATA 8DC0,BB,12,13,3E,5F,CD,5A,BB,5943  [9114]
540 DATA 8DC8,10,F4,C9,21,23,DE,CD,49,2D83  [4CE6]
541 DATA 8DD0,C6,21,0D,DE,CD,49,C6,CD,610D  [0830]
542 DATA 8DD8,DF,EF,CD,8D,EF,32,87,EF,4361  [F992]
543 DATA 8DE0,21,A4,AC,11,00,40,01,0C,2C1E  [1168]
544 DATA 8DE8,00,ED,B0,21,2F,DE,CD,49,2C83  [CD1E]
545 DATA 8DF0,C6,21,0D,DE,CD,49,C6,CD,610D  [1C3C]
546 DATA 8DF8,DF,EF,CD,8D,EF,32,8A,EF,437B  [14D6]
547 DATA 8E00,21,83,EF,11,00,30,01,0A,2C78  [AF2E]
548 DATA 8E08,00,ED,B0,3E,02,DD,21,00,2D86  [26A0]
549 DATA 8E10,30,DF,0E,EF,C3,00,EF,07,2731  [41D0]
550 DATA 8E18,30,04,30,00,00,40,00,A4,1EA4  [CCE8]
551 DATA 8E20,AC,06,00,7E,B7,28,04,23,5553  [6148]
552 DATA 8E28,04,18,F8,78,C9,21,89,CF,1B91  [A9B2]
553 DATA 8E30,CD,49,C6,CD,06,BB,FE,31,63C1  [D01A]
554 DATA 8E38,38,04,FE,37,38,05,CD,B0,014E  [60A6]
555 DATA 8E40,E7,18,F0,CD,5A,BB,D6,30,66F0  [E502]
556 DATA 8E48,21,4D,01,3D,28,1C,21,DE,019C  [A092]
557 DATA 8E50,00,3D,28,16,21,A7,00,3D,0889  [5D38]
558 DATA 8E58,28,10,21,85,00,3D,28,0A,1CDE  [5C78]
559 DATA 8E60,21,6F,00,3D,28,04,21,5F,09DD  [E87E]
560 DATA 8E68,00,3D,C3,68,BC,21,08,DE,140A  [8DAC]
561 DATA 8E70,CD,49,C6,21,A4,AC,36,00,69EC  [7CDE]
562 DATA 8E78,C3,3A,BD,21,FD,DD,CD,49,7FFF  [1AA2]
563 DATA 8E80,C6,21,40,00,ED,5B,2E,B6,65AE  [34D2]
564 DATA 8E88,3A,CD,B8,C3,9E,BC,00,00,3370  [BFDA]
565 DATA 8E90,00,00,00,00,97,32,37,B6,04A8  [3500]
566 DATA 8E98,DF,70,C6,3E,15,CD,5A,BB,6B33  [8B38]
567 DATA 8EA0,21,FF,3F,E5,D1,13,01,00,2036  [0E7A]
568 DATA 8EA8,10,36,00,ED,B0,11,FF,3F,0F55  [A2D6]
569 DATA 8EB0,D5,CD,9B,BC,E1,E5,97,11,44C3  [5A32]
570 DATA 8EB8,00,20,CB,BE,23,1B,BA,20,1AA0  [DBCE]
571 DATA 8EC0,F9,3E,06,CD,5A,BB,CD,6C,7EDA  [EAA4]
572 DATA 8EC8,BB,E1,23,3A,22,B6,B7,20,6086  [DDB4]
573 DATA 8ED0,28,11,A4,AC,06,00,C5,01,0FBB  [13A8]
574 DATA 8ED8,08,00,ED,B0,3E,2E,12,13,13DF  [69CC]
575 DATA 8EE0,01,03,00,ED,B0,23,23,23,0BF9  [1B70]
576 DATA 8EE8,C1,04,7E,D6,20,FE,3B,38,6196  [6EE6]
577 DATA 8EF0,E5,78,32,37,B6,21,A4,AC,6D60  [AAD0]
578 DATA 8EF8,C9,E5,21,02,00,11,12,4D,59ED  [B1A8]
579 DATA 8F00,CD,66,BB,E1,06,08,CD,F7,670D  [2E12]
580 DATA 8F08,C7,3E,2E,CD,5A,BB,06,03,6523  [A2D8]
581 DATA 8F10,CD,F7,C7,3E,20,CD,5A,BB,427B  [A636]
582 DATA 8F18,23,23,23,7E,D6,20,FE,3B,1D37  [F0A4]
583 DATA 8F20,38,E2,21,02,00,11,12,50,20B0  [CDD6]
584 DATA 8F28,CD,66,BB,11,A4,AC,21,01,6EA3  [46E8]
585 DATA 8F30,01,E5,CD,75,BB,CD,11,EF,2111  [BEE8]
586 DATA 8F38,CD,06,BB,E1,E5,F5,CD,75,7B63  [2246]
587 DATA 8F40,BB,CD,9C,BB,CD,11,EF,CD,71CF  [9EC6]
588 DATA 8F48,9C,BB,F1,E1,FE,E0,20,26,74E6  [9B24]
589 DATA 8F50,3A,37,B6,3C,32,37,B6,E5,0505  [F78C]
590 DATA 8F58,CD,75,BB,3E,16,CD,5A,BB,6CCB  [8388]
591 DATA 8F60,3E,01,CD,5A,BB,06,0C,CD,0655  [3CE6]
592 DATA 8F68,2A,EF,3E,16,CD,5A,BB,97,2E81  [FB40]
593 DATA 8F70,CD,5A,BB,E1,18,2F,FE,FC,680C  [AD72]
594 DATA 8F78,20,02,E1,C9,FE,F0,20,05,0445  [A780]
595 DATA 8F80,E5,C1,2D,18,2D,FE,F1,20,45B2  [ACE6]
596 DATA 8F88,05,E5,C1,2C,18,24,FE,F2,207E  [11E0]
597 DATA 8F90,20,0F,E5,C1,7C,FE,01,20,034A  [35B0]
598 DATA 8F98,03,3E,4F,2D,D6,0D,67,18,0362  [B7AA]
599 DATA 8FA0,11,FE,F3,20,20,E5,C1,7C,280A  [B6D2]
600 DATA 8FA8,FE,42,20,03,3E,F4,2C,C6,690E  [A7DA]
601 DATA 8FB0,0D,67,E5,CD,75,BB,CD,60,0F8E  [CB3E]
602 DATA 8FB8,BB,E1,C2,8D,F6,CD,B0,E7,7053  [0142]
603 DATA 8FC0,C5,E1,C3,8D,F6,FE,0D,C4,4EE6  [946C]
604 DATA 8FC8,B0,E7,C2,8D,F6,CD,6C,BB,75B7  [937A]
605 DATA 8FD0,21,A4,AC,3A,37,B6,47,C9,2C87  [04E0]
606 DATA 8FD8,CD,49,C6,21,7D,CF,CD,49,6F17  [7050]
607 DATA 8FE0,C6,CD,06,BB,D6,41,FE,02,5D7A  [DB3A]
608 DATA 8FE8,30,F7,47,C6,41,C3,5A,BB,204B  [2DE0]
609 DATA 8FF0,DF,6A,C6,21,F0,DD,CD,49,6AF7  [AA78]
610 DATA 8FF8,C6,21,38,D6,CD,49,C6,CD,672D  [C222]
611 DATA 9000,06,BB,06,00,FE,4B,CC,5A,2A1E  [E9BA]
612 DATA 9008,BB,28,2E,FE,44,C4,B0,E7,5D17  [0DDC]
613 DATA 9010,20,ED,CD,5A,BB,CD,52,C6,31CE  [6212]
614 DATA 9018,CD,52,C6,DF,70,C6,CD,D9,66EB  [A036]
615 DATA 9020,EF,CD,52,C6,06,00,E5,7E,4364  [1C98]
616 DATA 9028,B7,28,04,23,04,18,F8,E1,5261  [3222]
617 DATA 9030,CD,45,C7,21,F8,A6,E5,18,69BA  [F5CE]
618 DATA 9038,23,CD,52,C6,21,F0,D5,CD,274F  [B2C2]
619 DATA 9040,49,C6,CD,06,BB,CD,52,C6,0A4E  [06F6]
620 DATA 9048,CD,52,C6,21,8C,B8,E5,11,6F8B  [86C8]
621 DATA 9050,40,00,3E,2C,CD,A1,BC,E1,2075  [F880]
622 DATA 9058,E5,CD,52,C7,CD,7D,BC,E1,41F5  [B12A]
623 DATA 9060,CD,49,C6,3E,0A,CD,5A,BB,6C8B  [4434]
624 DATA 9068,CD,BD,C7,CD,06,BB,C3,F0,5E5A  [355A]
625 DATA 9070,C5,21,DF,F7,CD,49,C6,CD,785D  [F722]
626 DATA 9078,06,BB,FE,4A,CD,5A,BB,C2,3014  [AB08]
627 DATA 9080,F0,C5,C7,0C,57,69,72,6B,52F3  [679A]
628 DATA 9088,6C,69,63,68,20,28,4A,2F,27BB  [0088]
629 DATA 9090,4E,29,3F,07,20,00,00,00,2BD0  [930E]
630 DATA 90A0,00,00,00,00,CD,06,B9,21,0723  [DAE0]
631 DATA 90A8,82,D7,CD,49,C6,21,40,00,6FC4  [BC9C]
632 DATA 90B0,22,2C,B6,11,00,00,ED,53,0C59  [3042]
633 DATA 90B8,30,B6,1B,3E,16,CD,73,28,364A  [C096]
634 DATA 90C0,F5,D5,CD,D6,FE,D1,38,0C,5FC8  [6750]
635 DATA 90C8,B7,20,F6,31,00,C0,CD,09,4CC3  [BCA6]
636 DATA 90D0,B9,C3,F0,C5,21,B8,28,CD,7D65  [FCEA]
637 DATA 90D8,9D,28,D1,CD,61,28,FE,01,5025  [45AE]
638 DATA 90E0,28,17,B7,28,E6,21,78,FE,021A  [A18C]
639 DATA 90E8,CD,49,C6,CD,61,28,21,F0,63CA  [1CF2]
640 DATA 90F0,D5,CD,49,C6,CD,06,BB,18,5B9E  [6728]
641 DATA 90F8,AB,DD,E5,E1,11,44,00,ED,7105  [1DC6]
642 DATA 9100,52,22,2E,B6,3E,02,32,2B,2E97  [1840]
643 DATA 9108,B6,21,6A,FE,CD,49,C6,3A,571A  [07E2]
644 DATA 9110,CD,B8,CD,15,FF,CD,52,C6,54DE  [B244]
645 DATA 9118,CD,BD,C7,C3,09,B9,0A,0D,5FA5  [901E]
646 DATA 9120,4B,65,6E,6E,62,79,74,65,3599  [2562]
647 DATA 9128,3A,20,26,00,0A,0A,0D,18,11BA  [203A]
648 DATA 9130,52,65,61,64,20,65,72,72,3A22  [4EDE]
649 DATA 9138,6F,72,18,00,21,68,D7,CD,29CB  [7D9E]
650 DATA 9140,49,C6,3A,D8,A6,C6,41,CD,19A7  [5EC6]
651 DATA 9148,5A,BB,21,8E,D7,CD,49,C6,0AD8  [5C0E]
652 DATA 9150,CD,06,BB,21,F0,D5,CD,49,7777  [01CA]
653 DATA 9158,C6,CD,06,BB,CD,F0,FD,DF,5FBD  [7FAE]
654 DATA 9160,73,C6,CD,A1,C7,38,FB,C5,1C5B  [3002]
655 DATA 9168,E5,21,FD,DD,CD,49,C6,E1,6E91  [0F2A]
656 DATA 9170,E5,CD,49,C6,E1,C1,ED,5B,410D  [5D04]
657 DATA 9178,20,B6,CD,8C,BC,CD,92,C6,2BD6  [A336]
658 DATA 9180,D0,CD,5C,C6,21,A8,FE,C3,5E37  [7710]
659 DATA 9188,D4,DF,2E,55,CD,CD,29,D0,588E  [833C]
660 DATA 9190,11,00,00,62,CD,CD,29,D0,0B7E  [8B74]
661 DATA 9198,EB,06,00,09,EB,25,20,F4,73E8  [8F8A]
662 DATA 91A0,61,79,92,4F,9F,47,EB,09,3C4B  [65C2]
663 DATA 91A8,EB,CD,CD,29,D0,7A,CB,3F,5B31  [C75A]
664 DATA 91B0,CB,3F,8A,94,38,EA,91,38,7132  [28B2]
665 DATA 91B8,E7,7A,1F,8A,67,22,CE,B8,64D4  [1D08]
666 DATA 91C0,CD,B0,29,D0,32,CD,B8,37,4143  [DFB2]
667 DATA 91C8,C9,5F,07,07,07,07,E6,0F,7237  [9D8A]
668 DATA 91D0,CD,22,FF,7B,E6,0F,FE,0A,70AA  [1A5C]
669 DATA 91D8,38,02,C6,07,C6,30,C3,5A,031C  [AC7A]
670 DATA 91E0,BB,3A,22,B6,B7,C8,CD,52,5B70  [B4EC]
671 DATA 91E8,C6,CD,52,C6,21,24,D7,CD,569B  [D5F8]
672 DATA 91F0,49,C6,CD,06,BB,FE,4A,28,0A5C  [EC1E]
673 DATA 91F8,12,FE,4E,C4,B0,E7,20,F2,35AE  [E504]
674 DATA 9200,CD,5A,BB,CD,7D,BC,CD,92,6BA0  [166A]
675 DATA 9208,BC,37,C9,CD,5A,BB,3E,0B,467B  [E828]
676 DATA 9210,CD,5A,BB,C3,5A,BB,00,00,6B6C  [9FEE]
677 DATA 9228,02,00,00,00,00,00,00,00,0100  [E344]
678 DATA 9258,00,00,00,20,20,20,20,20,03E0  [718A]
679 DATA 9260,20,20,20,20,20,20,42,41,1F45  [43A8]
680 DATA 9268,43,4B,55,50,2D,4D,41,53,3C6D  [B068]
681 DATA 9270,54,45,52,20,20,20,63,6F,3229  [53D6]
682 DATA 9278,70,79,72,69,67,68,74,20,2CC0  [4F3E]
683 DATA 9280,31,39,38,36,20,20,62,79,139D  [03F6]
684 DATA 9288,20,41,4E,54,49,53,4F,46,0F1C  [2A5E]
685 DATA 9290,54,20,20,20,20,20,00,00,2580  [047E]
686 DATA *ENDE*                               [8BDC]
687 zeile=104:MEMORY &7FFF                   [FD24]
688 READ d$:IF d$="*ENDE*"THEN 700           [19A4]
689 adr=VAL("&"+d$)                          [E618]
690 pr=0                                     [4E1C]
691 FOR i=1 TO 8                             [0572]
692 READ a$:a=VAL("&"+a$)                    [2C50]
693 POKE adr,a:adr=adr+1                     [D92C]
694 pr=pr*2:IF pr>65535 THEN pr=pr-65535     [D6AC]
695 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                        [3DC4]
696 NEXT i                                   [3F1A]
697 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                           [59A6]
698 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                    [DD2E]
699 zeile=zeile+1:GOTO 688                   [B898]
700 SAVE"BACKUP.BIN",B,&8000,&1296:END       [F348]
```

**Listing 1. »Backup-Master« (Schluß)**

```
10 'BACKUP-MASTER                            [73AE]
20 'copyright 1986 by ANTISOFT               [BDFA]
30 'Gerd Weinand / Herrenstr. 14 / 5590
   Cochem                                    [0E44]
35 '                                         [7F5E]
40 MEMORY &7FFF                              [9E58]
50 LOAD"backup.bin",&8000                    [C1C0]
60 CALL &8000                                [5960]
```

**Listing 2. Diese Routine dient dem einfachen Aufruf des Backup-Master**

# Schrift beliebig groß

**Mit verschiedenen Schriftgrößen sah es bisher auf den CPC-Monitoren mager aus. Unsere Befehlserweiterung »Scale« schafft Ihnen jetzt freie Bahn.**

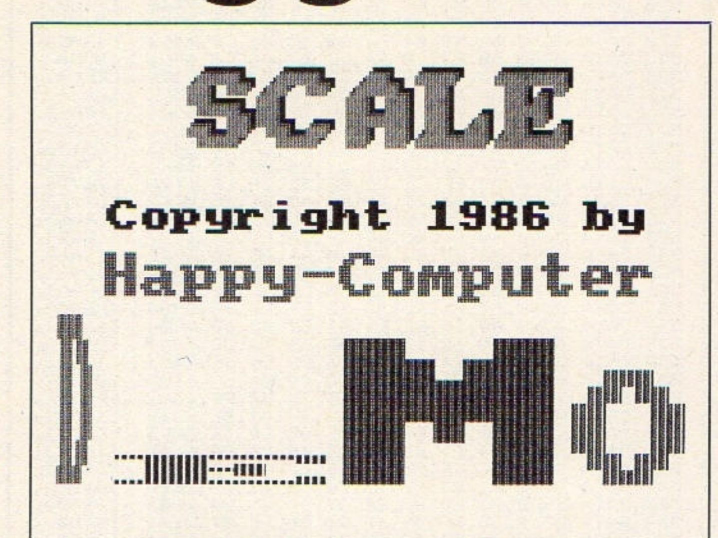
Vor allem Spiel- und Grafik-, aber auch Anwenderprogramme profitieren von einem ansprechenden Bildschirmaufbau. Sei es, um optische Reize zu erzeugen, oder auch »nur« für eine bessere Übersicht. Bisher stand jedoch kaum ein brauchbares Hilfsmittel zur Verfügung. Alles was es gab, war beispielsweise Software, die erlaubte, die drei Schriftgrößen der verschiedenen Bildschirmmodi zu verbinden.

Wir stellen Ihnen hier nun mit »Scale« eine RSX-Befehlserweiterung vor, die sich nach Belieben in jedes Ihrer eigenen Programme integrieren läßt. Sie besteht aus weniger als 200 Byte Maschinencode und ist im Speicher zwischen den Adressen 3FFF und AC00 hex frei verschiebbar. Listing 1 enthält den Basic-Lader. Speichern Sie ihn sicherheitshalber bitte unbedingt sofort nach der Eingabe. Nach dem Start erzeugt er selbsttätig die Binärdatei »SCALE.BIN«. Eine eindrucksvolle Demonstration finden Sie in Listing 2. Die Zeilen 10 bis 90 zeigen, wie man den Scale-Maschinencode in Programme einbindet und ihn dabei automatisch unterhalb der Speicher-Obergrenze ablegt. Die eigentliche Demonstration beginnt ab Zeile 100. Wie Sie auch dort sehen können, ist die Syntax des neuen Befehls

`|SCALE,x_größe,y_größe,"Text"`

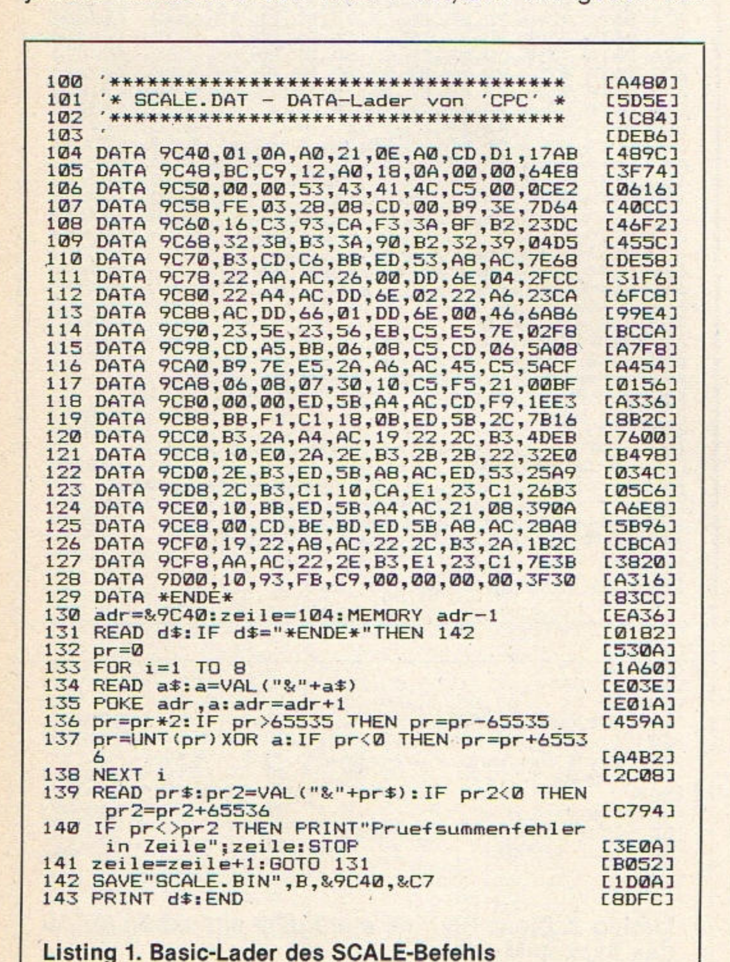
Die beiden Parameter »X_größe« und »Y_größe« müssen jeweils mindestens den Wert 1 besitzen und geben den Betrag der Vergrößerung in x- und y-Richtung an. »Text« steht für die auszugebenden ASCII-Zeichen. Die Koordinaten für die Ausgabe der vergrößerten Zeichen legen Sie durch Positionierung des Grafikcursors fest (MOVE). Freunde der Maschinensprache-Programmierung finden in Listing 3 den kompletten Quellcode.

Nun steht Ihrer Arbeit und dem Vergnügen mit Scale nichts mehr im Wege. Viel Spaß! (Gerd Weinand/ja)




So oder ähnlich kann in Zukunft auch Ihr Bildschirm aussehen, wenn Sie mit »Scale« arbeiten

| Steckbrief | |
|---|---|
| Programm: | Scale |
| Computer: | CPC 464 |
| Checksummer: | Explora/CPC |
| Datenträger: | Kassette/Diskette |



```
100 '***********************************   [A480]
101 '* SCALE.DAT - DATA-Lader von 'CPC' *  [5D5E]
102 '***********************************   [1C84]
103 '                                       [DEB6]
104 DATA 9C40,01,0A,A0,21,0E,A0,CD,D1,17AB [489C]
105 DATA 9C48,BC,C9,12,A0,18,0A,00,00,64E8 [3F74]
106 DATA 9C50,00,00,53,43,41,4C,C5,00,0CE2 [0616]
107 DATA 9C58,FE,03,28,08,CD,00,B9,3E,7D64 [40CC]
108 DATA 9C60,16,C3,93,CA,F3,3A,8F,B2,23DC [46F2]
109 DATA 9C68,32,38,B3,3A,90,B2,32,39,04D5 [455C]
110 DATA 9C70,B3,CD,C6,BB,ED,53,A8,AC,7E68 [DE58]
111 DATA 9C78,22,AA,AC,26,00,DD,6E,04,2FCC [31F6]
112 DATA 9C80,22,A4,AC,DD,6E,02,22,A6,23CA [6FC8]
113 DATA 9C88,AC,DD,66,01,DD,6E,00,46,6A86 [99E4]
114 DATA 9C90,23,5E,23,56,EB,C5,E5,7E,02F8 [BCCA]
115 DATA 9C98,CD,A5,BB,06,08,C5,CD,06,5A08 [A7F8]
116 DATA 9CA0,B9,7E,E5,2A,A6,AC,45,C5,5ACF [A454]
117 DATA 9CA8,06,08,07,30,10,C5,F5,21,00BF [0156]
118 DATA 9CB0,00,00,ED,5B,A4,AC,CD,F9,1EE3 [A336]
119 DATA 9CB8,BB,F1,C1,18,0B,ED,5B,2C,7B16 [8B2C]
120 DATA 9CC0,B3,2A,A4,AC,19,22,2C,B3,4DEB [7600]
121 DATA 9CC8,10,E0,2A,2E,B3,2B,2B,22,32E0 [B498]
122 DATA 9CD0,2E,B3,ED,5B,A8,AC,ED,53,25A9 [034C]
123 DATA 9CD8,2C,B3,C1,10,CA,E1,23,C1,26B3 [05C6]
124 DATA 9CE0,10,BB,ED,5B,A4,AC,21,08,390A [A6E8]
125 DATA 9CE8,00,CD,BE,BD,ED,5B,A8,AC,28A8 [5B96]
126 DATA 9CF0,19,22,A8,AC,22,2C,B3,2A,1B2C [CBCA]
127 DATA 9CF8,AA,AC,22,2E,B3,E1,23,C1,7E3B [3820]
128 DATA 9D00,10,93,FB,C9,00,00,00,00,3F30 [A316]
129 DATA *ENDE*                            [83CC]
130 adr=&9C40:zeile=104:MEMORY adr-1       [EA36]
131 READ d$:IF d$="*ENDE*"THEN 142         [0182]
132 pr=0                                   [530A]
133 FOR i=1 TO 8                           [1A60]
134 READ a$:a=VAL("&"+a$)                  [E03E]
135 POKE adr,a:adr=adr+1                   [E01A]
136 pr=pr*2:IF pr>65535 THEN pr=pr-65535   [459A]
137 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                      [A4B2]
138 NEXT i                                 [2C08]
139 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                         [C794]
140 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                  [3E0A]
141 zeile=zeile+1:GOTO 131                 [B052]
142 SAVE"SCALE.BIN",B,&9C40,&C7            [1D0A]
143 PRINT d$:END                           [8DFC]
```

Listing 1. Basic-Lader des SCALE-Befehls

```
10 adr=HIMEM-196:MEMORY adr-1              [F3F4]
20 LOAD"scale.bin",adr                     [9B70]
30 'Adressen anpassen                      [8FB0]
40 a=adr+18                                [48DA]
50 POKE adr+11,INT(a/256):POKE adr+10,a-
   256*INT(a/256)                          [57C6]
60 a=adr+14                                [4CD6]
70 POKE adr+5,INT(a/256):POKE adr+4,a-25
   6*INT(a/256)                            [5816]
80 a=adr+10:POKE adr+2,INT(a/256):POKE a
   dr+1,a-256*INT(a/256)                   [E142]
90 CALL adr:'Befehl einbinden              [82DE]
100 '                                      [DEB0]
110 ' *** SCALE-DEMO ***                   [80DE]
120 '                                      [E0B4]
130 MODE 1:INK 0,0:INK 1,26:INK 2,16:INK
     3,8:BORDER 0                          [721E]
140 PEN 3:a$="SCALE":MOVE 128,398:|SCALE
    ,10,5,@a$                              [A718]
150 PEN 2:a$="SCALE":MOVE 124,394:|SCALE
    ,10,5,@a$                              [A208]
160 PEN 1:a$="SCALE":MOVE 120,390:|SCALE
    ,10,5,@a$                              [24F8]
170 PEN 3:a$="Copyright 1986 by":MOVE 50
    ,270:|SCALE,4,2,@a$                    [5DC4]
180 PEN 2:a$="ANTISOFT":MOVE 164,220:|SC
    ALE,5,3,@a$                            [88A2]
190 POKE &B28F,&C3:a$="D":MOVE 0,160:|SC
    ALE,5,12,@a$                           [1EE0]
200 POKE &B28F,115:a$="E":MOVE 60,26:|SC
    ALE,30,2,@a$                           [6AD0]
```

Listing 2. Demonstration des Befehls SCALE

```
210 POKE &B28F,79:a$="M":MOVE 290,138:|S
    CALE,30,10,@a$                          [77C4]
220 POKE &B28F,202:a$="O":MOVE 520,108:|
    SCALE,16,8,@a$                          [2FBA]
230 PEN 1:CALL &BB18                        [DEE4]
240 FOR i=1 TO 26:PRINT CHR$(11);:NEXT      [965C]
250 PEN 1:a$="Anwendung:":MOVE 0,390:|SC
    ALE,6,4,@a$                             [D2CC]
260 PEN 2:a$="Der |SCALE - Befehl":MOVE
    0,290:|SCALE,4,2,@a$                    [16A8]
270 a$="vergroessert<3>alle":MOVE 0,230:
    |SCALE,4,2,@a$                          [D56E]
280 a$="Zeichen<3>stufenlos":MOVE 0,170:
    |SCALE,4,2,@a$                          [CC36]
290 a$="sowohl<2>in X-,<2>als":MOVE 0,11
    0:|SCALE,4,2,@a$                        [FB22]
300 a$="auch in Y-Richtung.":MOVE 0,50:|
    SCALE,4,2,@a$                           [F910]
310 CALL &BB18                              [5F06]
320 WINDOW#1,1,40,6,25:CLS#1                [5266]
330 a$="Er<2>benoetigt<2>drei":MOVE 0,29
    0:|SCALE,4,2,@a$                        [7A5A]
340 a$="Parameter:":MOVE 0,230:|SCALE,4,
    2,@a$                                   [C6CE]
350 PEN 3:a$="1. X-Vergroesserung":MOVE
    0,170:|SCALE,4,2,@a$                    [4FCA]
360 a$="2. Y-Vergroesserung":MOVE 0,110:
    |SCALE,4,2,@a$                          [61E4]
370 a$="3. Stringvariable":MOVE 0,50:|SC
    ALE,4,2,@a$                             [643A]
380 PEN 1:CALL &BB18                        [4DF0]
390 FOR i=1 TO 26:PRINT CHR$(11);:NEXT:M
    ODE 0:RANDOMIZE TIME                    [8E2A]
400 WHILE INKEY$=""                         [C1E4]
410   a$=CHR$(RND(1)*90+33)                 [8B26]
420   x=RND(1)*16+1:y=RND(1)*16+1           [CB82]
430   xm=RND(1)*639:ym=RND(1)*399           [01BA]
440   fa=RND(1)*14+1                        [079A]
450   PEN fa:MOVE xm,ym:|SCALE,x,y,a$       [16C6]
460 WEND                                    [37D0]
470 FOR i=1 TO 26:PRINT CHR$(11);:NEXT:M
    ODE 2:PEN 1                             [2BA4]
```

**Listing 2. Demonstration des Befehls SCALE (Schluß)**

```
        ORG     &A000       ; Initialisierung ab A000 hex
                            ; RSX-Befehl >>SCALE<< Version vom 15.04.86
                            ; von Gerd Weinand, Herrenstr. 14, 5590 Cochem
;
                            ; erster  Parameter: X-Größe
                            ; zweiter Parameter: Y-Größe
                            ; dritter Parameter: Stringvariable
;
LOGEXT  EQU     &BCD1
XGROSS  EQU     &ACA4
YGROSS  EQU     &ACA6
XPOS    EQU     &ACA8
YPOS    EQU     &ACAA
GETMTX  EQU     &BBA5
DRAWR   EQU     &BBF9
MOVE    EQU     &BBC0
ASKCUR  EQU     &BBC6
INIT:   LD      BC,RSX      ; Adresse der Befehlstabelle
        LD      HL,KERNAL   ; vier freie Byte Für Betriebssystem
        CALL    LOGEXT      ; Befehl einbinden
        RET                 ; zurück ins Basic
RSX:    DW      TABLE       ; Adresse des Befehlsnamens
        JR      START       ; Aufruf der Befehlsroutine
KERNAL: DS      4           ; vier Byte Für Betriebssystem
TABLE:  DM      "SCAL"      ; Befehlsname in Großbuchstaben
        DB      &C5         ; "E" +80 hex (letzter Buchstabe)
        DB      0           ; Ende der Befehlsnamens-Tabelle
;
START:  CP      3           ; drei Parameter?
        JR      Z,PARAM     ; wenn ja: weiter bei PARAM
        CALL    &B900       ; Basic-Interpreter einschalten
        LD      A,&16       ; Fehlercode >>OPERAND MISSING<<
        JP      &CA93       ; Fehlermeldung ausgeben und ins Basic
;
PARAM:  DI                  ; Interrupts sperren
        LD      A,(&B28F)   ; Text-Farbstift nach A
        LD      (&B338),A
        LD      A,(&B290)   ; Text-Paper nach A
        LD      (&B339),A
        CALL    ASKCUR      ; Grafikcursor-Position laden
        LD      (XPOS),DE   ; X-Position zwischenspeichern
        LD      (YPOS),HL   ; Y-Position zwischenspeichern
        LD      H,0
        LD      L,(IX+4)    ; X-Größe nach L
        LD      (XGROSS),HL
        LD      L,(IX+2)    ; Y-Größe nach L
        LD      (YGROSS),HL
        LD      H,(IX+1)
        LD      L,(IX+0)    ; Stringdescriptor in HL speichern
        LD      B,(HL)      ; Stringlänge in B speichern
        INC     HL
        LD      E,(HL)      ; Stringadresse (Low-Byte)
        INC     HL
        LD      D,(HL)      ; Stringadresse (High-Byte)
        EX      DE,HL       ; mit HL vertauschen: Stringadresse in HL
;
LETTER: PUSH    BC          ; Stringlänge merken
        PUSH    HL          ; Stringadresse merken
        LD      A,(HL)      ; Zeichen nach A
        CALL    GETMTX      ; Matrixadresse des Zeichens nach HL
        LD      B,8         ; Höhe des Zeichens
;
HOCH:   PUSH    BC          ; Höhe des Zeichens merken
        CALL    &B906       ; Betriebssystem-ROM einschalten
        LD      A,(HL)      ; Zeichenmatrix-Reihe nach A
        PUSH    HL          ; Matrixadresse des Zeichens merken
        LD      HL,(YGROSS) ; Y-Größe
        LD      B,L
LOOP:   PUSH    BC          ; Y-Größe merken
        LD      B,8         ; Breite des Zeichens
;
BREIT:  RLCA                ; Akku links rotieren; achtes Bit nach CARRY
        JR      NC,NOINK    ; kein Punkt: weiter bei NOINK
        PUSH    BC          ; Breite des Zeichens merken
        PUSH    AF          ; Matrixreihe im Akku merken
        LD      HL,0        ; Y-Offset
        LD      DE,(XGROSS) ; X-Offset
        CALL    DRAWR       ; Linie relativ ziehen
        POP     AF          ; Matrixreihe
        POP     BC          ; Breite des Zeichens
        JR      WEITER      ; NOINK überspringen
NOINK:  LD      DE,(&B32C)  ; laufende X-Ordinate
        LD      HL,(XGROSS) ; X-Offset
        ADD     HL,DE       ; Addieren
        LD      (&B32C),HL  ; neue X-Ordinate speichern
WEITER: DJNZ    BREIT
;
        LD      HL,(&B32E)  ; laufende Y-Ordinate
        DEC     HL
        DEC     HL          ; minus 2
        LD      (&B32E),HL  ; neue Y-Ordinate speichern
        LD      DE,(XPOS)   ; linker Rand des Zeichens
        LD      (&B32C),DE  ; neue X-Ordinate speichern
        POP     BC          ; Y-Größe
        DJNZ    LOOP
        POP     HL          ; Matrixadresse
        INC     HL          ; nächste Matrixreihe
        POP     BC          ; Höhe des Zeichens
        DJNZ    HOCH
;
        LD      DE,(XGROSS) ; X-Größe
        LD      HL,&0008    ; *8= linker Rand nächstes Zeichen
        CALL    &BDBE       ; vorzeichenlose Multiplikation
        LD      DE,(XPOS)   ; absolute Ordinate linker Rand
        ADD     HL,DE       ; abs. und rel. Ordinaten addieren
        LD      (XPOS),HL   ; in XPOS speichern und
        LD      (&B32C),HL  ; als laufende X-Ordinate speichern
        LD      HL,(YPOS)
        LD      (&B32E),HL  ; neue Y-Ordinate
        POP     HL          ; Stringadresse
        INC     HL          ; nächstes Zeichen
        POP     BC          ; Stringlänge
        DJNZ    LETTER
        EI                  ; Interrupts wieder zulassen
        RET                 ; Fertig
```

**Listing 3. Der Assembler-Quellcode zeigt die Arbeitsweise**

# Programme in der Zange

**Oft wünscht man sich, Dateien würden weniger Kapazität des Speichermediums beanspruchen. »Kompex« macht diese Träume wahr.**

Verschwendung von Speicherplatz ist nicht nur teuer, sie macht die Arbeit auch unübersichtlicher. Wenn Sie nun beispielsweise Ihre 20 Programme auf nur noch zwei Disketten (oder auch Kassetten) unterbekommen, anstatt wie bisher mit derselben Software derer vier zu belegen, wäre das nicht die Erlösung? Kein träumerisches Wunschdenken, sondern ein durchaus realisierbares Vorhaben, wie unser Programm-Paket »Kompex« mit Bravour unter Beweis stellt. Dieser Name beschreibt die Verknüpfung der Funktionen beider Programmteile. Die Silbe »Komp« steht dabei für »komprimieren«. Die nötige Rückwandlung der gestauchten Daten übernimmt dann der »Expander« (deshalb »ex«). Wie funktioniert nun dieses praktische Utility?

Kompex untersucht den zu komprimierenden Speicherbereich auf Byte-Wiederholungen. Einzelne Bytes erfahren keine Veränderung, während aufeinanderfolgende gleiche Werte zu jeweils drei Byte zusammengesetzt werden, die den Wert und die Anzahl der Wiederholungen enthalten. Im theoretischen Fall, daß der Quell-Speicherbereich nur aus Byte-Pärchen besteht, würde dies einen Zuwachs von 50 Prozent bedeuten. In der Praxis aber tritt der Fall einer Vergrößerung nur äußerst selten ein. Meist ist mit einer Ersparnis von bis zu 50 Prozent zu rechnen.

Während der Komprimierung (und natürlich auch bei der späteren Expansion) steht sowohl der Quell- als auch der Zielcode im Arbeitsspeicher. Die maximale Länge zur Bearbeitung ist dadurch natürlich beschränkt. Je nach Beschaffenheit sind jedoch normalerweise auch Speicherbereiche von weit mehr als 20 KByte zu verarbeiten. Eine optimale Nutzung des verfügbaren Speicherraums ergibt sich, wenn Sie den Quellcode an eine möglichst hohe Adresse laden und die Anfangsadresse für den komprimierten Code an eine niedrige. Der wachsende Zielcode darf ruhig die unteren Byte des Quellcodes überschreiben, sofern dieser Teil bereits verarbeitet ist. Ist das nicht gewährleistet, bricht Kompex mit der Meldung »BREAK« ab. Diese Überprüfung arbeitet nur dann korrekt, wenn der Quellcode oberhalb des Zielcodes liegt.

Nach der Eingabe der Basic-Lader speichern Sie sie bitte zunächst sicherheitshalber. Nach dem Starten erzeugen beide jeweils eine Binärdatei mit Maschinencode. Listing 1 speichert den Kompressor »KOMPEX.KMP«, den Expander enthält »KOMPEX.EXP« in Listing 2.

Eine stellvertretende Befehlsfolge zur Benutzung des Kompressors ist

```
10 MEMORY adresse-1
20 LOAD "KOMPEX.KMP",adresse
30 l%=0
40 LOAD "QUELL",anfang
50 CALL adresse,anfang,länge,zieladresse,@l%
60 SAVE "ZIEL",b,zieladresse,l%
```

Zeile 10 reserviert dem Maschinencode des Kompressors Speicherplatz. Der Wert für »adresse« ist variabel, da Kompex frei verschiebbar ist. Nach dem Laden definiert Zeile 30 eine Integervariable, die später Bedeutung erhält. Das Quellprogramm lädt dann Zeile 40; »anfang« steht für dessen Ladeadresse. Der Aufruf in Zeile 50 benötigt als Parameter auch die Länge des Quellcodes (»länge«), die Adresse, ab der der Zielcode abzulegen ist (»zieladresse«), und »l%« für die Rückmeldung der Länge des komprimierten Endprodukts. Die beiden letztgenannten Werte benötigen wir auch wieder in Zeile

```
100 '*************************************  [31D4]
101 '*KOMPEXK.DAT - DATA-Lader von 'CPC' *  [5B8C]
102 '*************************************  [A3D8]
103 '                                       [DEB6]
104 DATA A500,FE,04,C0,DD,6E,00,DD,66,697C  [9DE6]
105 DATA A508,01,E5,DD,6E,02,DD,66,03,272B  [7F9C]
106 DATA A510,E5,DD,6E,04,DD,66,05,E5,4FDF  [F018]
107 DATA A518,DD,6E,06,DD,66,07,C1,FD,7A43  [24FC]
108 DATA A520,E1,FD,E5,E5,21,00,00,ED,5CD5  [13D0]
109 DATA A528,42,E5,C1,AF,B8,20,07,B9,0F67  [F9B8]
110 DATA A530,20,04,E1,E1,E1,C9,E1,7E,06A0  [DA7C]
111 DATA A538,FD,77,00,23,FD,23,0C,20,662C  [0B78]
112 DATA A540,03,04,28,3E,57,7E,FD,77,04AD  [768C]
113 DATA A548,00,23,FD,23,0C,20,03,04,15B2  [9F0E]
114 DATA A550,28,30,BA,20,EF,FD,36,00,09A0  [A482]
115 DATA A558,00,FD,7D,BD,20,0B,FD,E5,3B03  [E7F6]
116 DATA A560,FD,6C,FD,7D,FD,E1,BC,28,78CC  [F08C]
117 DATA A568,29,7E,BA,20,11,FD,34,00,1D54  [2B80]
118 DATA A570,23,0C,20,03,04,28,0B,3E,1618  [DE04]
119 DATA A578,FF,FD,BE,00,20,EB,FD,23,5475  [B820]
120 DATA A580,18,B5,FD,E5,E1,23,D1,ED,367B  [BEEC]
121 DATA A588,52,DD,E1,DD,75,00,DD,74,0DD6  [19F4]
122 DATA A590,01,C9,CD,00,B9,CD,6B,CB,2D81  [3FF8]
123 DATA *ENDE*                             [87C0]
124 adr=&A500:zeile=104:MEMORY adr-1        [8028]
125 READ d$:IF d$="*ENDE*"THEN 136          [1C8E]
126 pr=0                                    [4910]
127 FOR i=1 TO 8                            [0066]
128 READ a$:a=VAL("&"+a$)                   [BF44]
129 POKE adr,a:adr=adr+1                    [5020]
130 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [FF8E]
131 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                       [12A6]
132 NEXT i                                  [2DFC]
133 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                          [8388]
134 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                   [2E10]
135 zeile=zeile+1:GOTO 125                  [B95E]
136 SAVE"KOMPEX.KMP",B,&A500,&98:END        [7C02]
```

**Listing 1. Mit dem Kompressor bringen Sie Ihre Programme in ein platzsparendes Format**

```
100 '*************************************  [31D4]
101 '*KOMPEXE.DAT - DATA-Lader von 'CPC' *  [7980]
102 '*************************************  [A3D8]
103 '                                       [DEB6]
104 DATA A500,FE,04,C0,DD,6E,00,DD,66,697C  [9DE6]
105 DATA A508,01,E5,DD,6E,02,DD,66,03,272B  [7F9C]
106 DATA A510,E5,DD,6E,04,DD,66,05,E5,4FDF  [F018]
107 DATA A518,DD,6E,06,DD,66,07,C1,FD,7A43  [24FC]
108 DATA A520,E1,E5,21,00,00,ED,42,E5,4E35  [1568]
109 DATA A528,C1,E1,AF,B9,20,05,B8,20,46F4  [5BA6]
110 DATA A530,02,E1,C9,7E,FD,77,00,0C,21B8  [CA92]
111 DATA A538,20,03,04,28,3D,23,FD,23,127D  [262C]
112 DATA A540,57,7E,FD,77,00,0C,20,03,2CA3  [756A]
113 DATA A548,04,28,2F,23,FD,23,BA,20,09E0  [9374]
114 DATA A550,EF,57,7E,FE,00,28,1C,5F,62A7  [7FDC]
115 DATA A558,7A,FD,77,00,FD,23,F5,FD,0AD3  [4418]
116 DATA A560,7D,BD,20,0B,FD,E5,FD,6C,109A  [6D22]
117 DATA A568,FD,7D,FD,E1,BC,28,13,F1,75E7  [2A34]
118 DATA A570,1D,20,E6,23,0C,20,BC,04,19EC  [AC88]
119 DATA A578,20,B9,E1,AF,BD,20,02,BC,2D40  [3BD2]
120 DATA A580,C8,E9,CD,00,B9,CD,6B,CB,4101  [3AF8]
121 DATA A588,52,DD,E1,DD,75,00,DD,74,0DD6  [19F4]
122 DATA A590,01,C9,CD,00,B9,CD,6B,CB,2D81  [3FF8]
124 DATA *ENDE*                             [4BC2]
125 adr=&A500:zeile=104:MEMORY adr-1        [CF2A]
126 READ d$:IF d$="*ENDE*"THEN 137          [0592]
127 pr=0                                    [5212]
128 FOR i=1 TO 8                            [0F68]
129 READ a$:a=VAL("&"+a$)                   [F546]
130 POKE adr,a:adr=adr+1                    [B510]
131 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [0490]
132 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                       [A1A8]
133 NEXT i                                  [1CFE]
134 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                          [A78A]
135 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                   [3B12]
136 zeile=zeile+1:GOTO 126                  [2262]
137 SAVE"KOMPEX.EXP",B,&A500,&98:END        [AD0E]
```

**Listing 2. Vor der Nutzung der komprimierten Programme müssen Sie sie wieder expandieren**

60 zum Speichern des Zielcodes. Merken Sie sich bitte den Wert der Variablen »l%« für die Expansion. Sie ist nötig, da ein gestauchtes Programm natürlich in dieser Form nicht funktionsfähig ist.

Den Expander laden Sie beispielsweise mit

```
10 MEMORY adresse-1
20 LOAD "KOMPEX.EXP",adresse
30 LOAD "ZIEL",anfang
40 CALL adresse,anfang,länge,zieladresse,start
```

Laden Sie das codierte Programm soweit wie möglich an hohe Speicheradressen. Wenn Kompex dann bei der Expansion noch nicht gelesene Bytes zu überschreiben droht, bricht er auch hier ab. Wählen Sie für <start> anstelle einer Adresse die Null, kehrt der Expander ins Basic zurück.

Wenn Sie nun jedes gespeicherte Programm mit einem Lader versehen, der Kompex enthält, läßt es sich automatisch beim Laden mit »RUN "LADERNAME"« expandieren.

(Andreas Illenseer/ja)

| Steckbrief | |
|---|---|
| Programm: | Kompex |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora/CPC |
| Datenträger: | Kassette/Diskette |

# Das Disketten-Plus

**Viele Fähigkeiten der 3-Zoll-Diskettenlaufwerke unterstützt das Amsdos nicht. »Disk Plus« macht sie aber jetzt sogar aus dem Basic zugänglich.**

Das Disketten-Betriebssystem der CPC-Serie – Amsdos genannt – ist mit einer tückischen Eigenart behaftet. Es bringt während des Programmlaufs meist an den unpassendsten Stellen seine Fehlermeldungen aufs Tapet. Da diese nicht zu unterdrücken sind, kehrt es dann zum READY-Modus des Basic-Interpreters zurück und bricht so den Programmlauf ab.

»Disk Plus« ist eine RSX-Basic-Erweiterung und ergänzt die Basic-Befehle DIR, ERA und REN. Sie entlockt dem DOS Informationen, die sonst aus dem Basic nicht zugänglich sind.

Im einzelnen stehen folgende neue Befehle zur Verfügung:

**GETDIR,**@name$,@a$(0)
liest das Directory der Diskette in die Stringdimension a$(). Dabei hält er die alphabetische Reihenfolge ein und duldet nur Einträge, die mit der Maske »name$« übereinstimmen (wie unter CP/M: »*.*«, »B?.*«, »*.BAS« und so weiter). Die Stringdimension müssen Sie ausreichend groß dimensionieren und jeden String mit zwölf Zeichen vorbesetzen. Dazu dient beispielsweise der Befehl »a$(0)=space$(12)«.

**TESTDRIVE,**@status%
übergibt in der Variablen »status%« den Wert Null, wenn er im Laufwerk keine Diskette findet. Ist eine Diskette eingelegt, enthält die Variable den Wert –1.

**TESTNAME,**@name$,@status%
unterdrückt den »Bad Command Error«, der sich immer dann meldet, wenn der Benutzer dem DOS einen unzulässigen Dateinamen vorsetzt. Das Auftreten dieses Fehlers zeigt eine Null in der Variablen »status%« an.

**TESTFILE,**@name$,@status%
prüft, ob der in »name$« angegebene Dateiname auf der Diskette vorhanden ist, und meldet dessen Existenz mit dem Wert –1 in der Variablen »status%«.

**SET.RO,**@name$
**SET.RW,**@name$
**SET.SYS,**@name$
**SET.DIR,**@name$
Das DOS kann nicht nur Dateien löschen oder umbenennen, sondern auch schützen oder verstecken. Bisher war das nur über CP/M möglich. SET erledigt nun diese Arbeit. Mit dem Zusatz »RO« (Read Only) schützt er den Eintrag »name$« vor dem Löschen, »RW« (Read-Write) macht diese Veränderung wieder ungeschehen. »SYS« (System) versteckt Dateien vor den Augen des Benutzers. Sie erscheinen im Directory einfach nicht mehr. »DIR« (Directory) bringt sie wieder ans Tageslicht.

**ATTRIBUT,**@name$,@rostatus%,@sysstatus%
gibt den Status der Datei »name$« aus, der mit dem vorangegangenen Befehls-Quartett festzusetzen ist. Zwei Variablen zeigen den Zustand des Eintrags. »rostatus%« ist Null, wenn der Eintrag zum Löschen freigegeben ist, –1 signalisiert den Schutz. »sysstatus%« bekundet mit dem Wert –1, daß »name$« versteckt ist, und mit einer Null das Gegenteil.

Nach der Eingabe des Basic-Laders (Listing 1) speichern Sie ihn bitte erst einmal zur Sicherheit. Nach dem Start erzeugt er auf der Diskette die Binärdatei »DISK+.BIN« mit dem Maschinencode der Befehlserweiterung. Wie bei allen RSX-Befehlen beginnen auch diese mit dem senkrechten Strich, den Sie durch Druck der Tasten <SHIFT> und <@> erhalten. Die Befehle aktivieren Sie durch

```
MEMORY &9FFF
LOAD "DISK+.BIN"
CALL &A000
```

Eine Anwendung zeigt Listing 2, das den Einsatz des Befehls GETDIR demonstriert.

(Stefan Aust/ja)

| Steckbrief | |
|---|---|
| Programm: | Disk Plus |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora/CPC |
| Datenträger: | Diskette |

```
100 '************************************   [A480]
101 '* DISK+.DAT - DATA-Lader von 'CPC' *   [D03A]
102 '************************************   [1C84]
103 '                                        [DEB6]
104 DATA A000,01,09,A0,21,68,A0,C3,D1,1447  [C808]
105 DATA A008,BC,26,A0,C3,6C,A0,C3,5A,4F8C  [47E8]
106 DATA A010,A1,C3,7C,A1,C3,A9,A1,C3,60ED  [9FD4]
107 DATA A018,DC,A1,C3,E9,A1,C3,F6,A1,57F9  [010C]
108 DATA A020,C3,04,A2,C3,32,A2,47,45,7B23  [263E]
109 DATA A028,54,44,49,D2,54,45,53,54,3C46  [1C24]
110 DATA A030,44,52,49,56,C5,54,45,53,3D61  [AF06]
111 DATA A038,54,4E,41,4D,C5,54,45,53,32D1  [3040]
112 DATA A040,54,46,49,4C,C5,53,45,54,31DA  [F142]
113 DATA A048,2E,52,CF,53,45,54,2E,52,1C26  [9E66]
114 DATA A050,D7,53,45,54,2E,44,49,D2,7280  [3434]
115 DATA A058,53,45,54,2E,53,59,D3,41,32BB  [ED4E]
116 DATA A060,54,54,52,49,42,55,D4,00,333C  [7DFE]
117 DATA A068,00,00,00,00,DF,70,A0,C9,06B1  [7F2A]
118 DATA A070,73,A0,07,CD,73,CD,CD,C1,1C47  [77D8]
119 DATA A078,CD,CD,CF,CD,2B,2B,7E,32,41CA  [A462]
120 DATA A080,57,A1,23,23,E5,CD,C7,CD,00CF  [7ECA]
```

**Listing 1. Der Basic-Lader für die Befehlserweiterung »Disk Plus«**

```
121 DATA A088,22,58,A1,CD,A6,DA,CD,14,1826  [40BC]
122 DATA A090,CE,CD,83,D6,DD,E1,AF,F5,4D87  [4760]
123 DATA A098,CD,98,D6,30,06,F1,CD,AF,5B81  [09FC]
124 DATA A0A0,A0,18,F4,F1,2A,58,A1,77,4695  [F384]
125 DATA A0A8,CD,F9,A0,CD,1E,A1,C9,C5,4333  [F61E]
126 DATA A0B0,E5,DD,E5,4F,06,00,CD,39,5C03  [69D8]
127 DATA A0B8,A1,79,B8,28,32,CD,23,D6,5974  [5CAA]
128 DATA A0C0,30,0C,04,CD,2B,A1,3A,57,14AF  [25B0]
129 DATA A0C8,A1,B8,20,EA,18,25,D5,3A,75E4  [F6C8]
130 DATA A0D0,57,A1,B9,20,01,0D,CD,4E,1708  [F37E]
131 DATA A0D8,A1,CD,39,A1,B8,28,0F,CD,6B43  [44FE]
132 DATA A0E0,32,A1,EB,CD,39,A1,EB,CD,22A7  [801E]
133 DATA A0E8,40,A1,EB,3D,18,EE,D1,0C,1426  [47CE]
134 DATA A0F0,CD,40,A1,79,DD,E1,E1,C1,615F  [F5F8]
135 DATA A0F8,C9,B7,C8,F5,4F,CD,39,A1,5E8F  [9760]
136 DATA A100,54,5D,13,06,08,1A,CB,BF,3E41  [3484]
137 DATA A108,77,13,23,10,F8,36,2E,06,3D62  [8D2C]
138 DATA A110,03,23,CB,BE,10,FB,CD,2B,191D  [72D8]
139 DATA A118,A1,0D,20,E1,F1,C9,4F,3A,5DD8  [C5E4]
140 DATA A120,57,A1,B9,C8,DD,36,00,00,1E50  [455E]
141 DATA A128,0C,18,F7,DD,23,DD,23,DD,11C7  [06DE]
142 DATA A130,23,C9,DD,2B,DD,2B,DD,2B,3D05  [A806]
143 DATA A138,C9,DD,6E,01,DD,66,02,C9,59AD  [2400]
144 DATA A140,C5,D5,E5,EB,01,0C,00,CD,4525  [05A4]
145 DATA A148,1B,B9,E1,D1,C1,C9,78,B9,3795  [18D2]
146 DATA A150,C8,3C,CD,2B,A1,18,F8,00,7488  [06B0]
147 DATA A158,00,00,DF,5E,A1,C9,61,A1,184F  [719E]
148 DATA A160,07,CD,73,CD,CD,C2,CD,CD,3647  [DC30]
149 DATA A168,CF,CD,FD,7E,00,E5,CD,30,4EBE  [617C]
150 DATA A170,C6,E1,30,05,CB,6F,C2,D0,5BA0  [79BC]
151 DATA A178,A1,C3,D6,A1,DF,80,A1,C9,75E3  [D9EC]
152 DATA A180,83,A1,07,CD,73,CD,CD,C1,6407  [BDC0]
153 DATA A188,CD,CD,CF,CD,E5,CD,C7,CD,45AF  [EAE2]
154 DATA A190,11,E4,00,CD,98,CA,D5,CD,3BDF  [E622]
155 DATA A198,ED,DA,D1,E1,30,08,13,13,55A5  [D0A0]
156 DATA A1A0,1A,FE,20,C2,D0,A1,C3,D6,3FF4  [6B04]
157 DATA A1A8,A1,DF,AD,A1,C9,B0,A1,07,7D3D  [E134]
158 DATA A1B0,CD,73,CD,CD,C1,CD,CD,CF,6B59  [7CB0]
159 DATA A1B8,CD,E5,CD,C7,CD,CD,5B,DA,4F20  [A4B2]
160 DATA A1C0,CD,14,CE,CD,83,D6,CD,98,70D2  [9E2E]
161 DATA A1C8,D6,E1,DA,D0,A1,C3,D6,A1,4209  [BAEE]
162 DATA A1D0,36,FF,23,36,FF,C9,36,00,2770  [2592]
163 DATA A1D8,23,36,00,C9,DF,E0,A1,C9,1463  [52A6]
164 DATA A1E0,E3,A1,07,CD,12,A2,CB,FE,5680  [5FF2]
165 DATA A1E8,C9,DF,ED,A1,C9,F0,A1,07,403D  [3630]
166 DATA A1F0,CD,12,A2,CB,BE,C9,DF,FA,7D60  [4584]
167 DATA A1F8,A1,C9,FD,A1,07,CD,12,A2,74FA  [7228]
168 DATA A200,23,CB,BE,C9,DF,08,A2,C9,3F45  [8710]
169 DATA A208,0B,A2,07,CD,12,A2,23,CB,23A5  [D99E]
170 DATA A210,FE,C9,CD,77,CD,CD,C2,CD,5785  [F162]
171 DATA A218,CD,5F,A2,E3,E5,CD,2C,A2,6FD6  [1C4E]
172 DATA A220,E1,E3,CD,7A,D9,CD,98,D6,525A  [E41C]
173 DATA A228,38,F1,E1,C9,E5,21,09,00,374E  [5876]
174 DATA A230,19,C9,DF,36,A2,C9,39,A2,20A4  [1CAC]
175 DATA A238,07,CD,73,CD,3D,CD,C1,CD,31E3  [292A]
176 DATA A240,CD,CF,CD,E5,CD,CF,CD,E5,469B  [FAC4]
177 DATA A248,CD,5F,A2,CD,D9,D9,E1,DC,6D62  [2B72]
178 DATA A250,D0,A1,D4,D6,A1,CD,DF,D9,50FB  [2B46]
179 DATA A258,E1,DA,D0,A1,C3,D6,A1,CD,52DF  [6C44]
180 DATA A260,C7,CD,CD,A6,DA,CD,14,CE,4602  [4538]
181 DATA A268,CD,83,D6,CD,98,D6,D2,0C,5660  [72F2]
182 DATA A270,D5,C9,00,00,00,00,00,00,58C0  [AEE0]
183 DATA *ENDE*                            [91CC]
184 adr=&A000:zeile=104:MEMORY adr-1      [AB2A]
185 READ d$:IF d$="*ENDE*"THEN 196        [AEA6]
186 pr=0                                  [6D1C]
187 FOR i=1 TO 8                          [BC72]
188 READ a$:a=VAL("&"+a$)                 [A950]
189 POKE adr,a:adr=adr+1                  [BC2C]
190 pr=pr*2:IF pr>65535 THEN pr=pr-65535  [679A]
191 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                     [EEB2]
192 NEXT i                                [3208]
193 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                        [CD94]
194 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
     in Zeile";zeile:STOP                 [D21C]
195 zeile=zeile+1:GOTO 185                [C376]
196 SAVE"DISK+.BIN",B,&A000,&271:END      [933C]
197 PRINT d$:END                          [460E]
```

**Listing 1. Der Basic-Lader für »Disk Plus« (Schluß)**

```
10 '***************************          [B92C]
20 '* GETDIR.BAS - DISK+ Demo *          [7DD0]
30 '***************************          [3130]
40 IF HIMEM=40959 THEN 60                [5180]
50 MEMORY &9FFF:LOAD"!DISK+.BIN":CALL &A
   000                                   [8070]
60 DIM a$(63):st%=0                      [6AE4]
70 FOR i=0 TO 63:a$(i)=SPACE$(12):NEXT   [1346]
80 MODE 1:PAPER 0:PEN 1:PRINT"BITTE DISK
   ETTE EINLEGEN":PRINT"UND EINE TASTE D
   RUECKEN"                              [4434]
90 WHILE INKEY$<>"":WEND:CALL &BB06      [64E4]
100 |TESTDRIVE,@st%                      [49BE]
110 IF st%=0 THEN PRINT"GG";:GOTO 80     [C4BC]
120 PRINT:INPUT"Suchmaske fuer DIR:_",na
    me$                                  [1E9A]
130 |TESTNAME,@name$,@st%                [2374]
140 IF st%=0 THEN 120                    [0FBE]
150 |GETDIR,@name$,@a$(0)                [7028]
160 l=ASC(name$)-1:f=1                   [3972]
170 FOR i=0 TO l                         [78C8]
180 FOR j=1 TO LEN(a$(i))                [C1D4]
190 PEN f:f=f+1+(f=3)*3                  [E4E2]
200 PRINT MID$(a$(i),j,1);:NEXT          [2A28]
210 PRINT:f=f+1+(f=3)*3:NEXT             [CC0E]
220 PEN 1:PRINT:PRINT"Ende (J/N):";:CALL
     &BB8A                               [62F8]
230 WHILE INKEY$<>"":WEND                [9030]
240 a$=UPPER$(INKEY$):IF a$<>"J"AND a$<>
    "N"AND a$<>CHR$(13)THEN 240          [B548]
250 PRINT a$:IF a$<>"J"THEN 70           [402A]
260 END                                  [B71E]
```

**Listing 2. Die Demonstration zeigt eine Anwendung des Befehls GETDIR**

# Zahlenumwandlung

**Computer können eines ganz besonders gut: mit Zahlen umgehen. Oft ergibt sich in Programmen auch die Notwendigkeit, Zahlen in Worten auszugeben. Mit »Ziffwort« ist das kein Problem mehr.**

Nehmen Sie an, Sie wollten ein Programm schreiben, das beispielsweise automatisch Überweisungsformulare ausfüllt. Dann stehen Sie vor einem großen Problem: Wie bringen Sie Ihrem Computer die korrekte Schreibweise für Zahlen bei? Diese Aufgabe erledigt nun »Ziffwort«, das sich als Unterroutine in eigene Programme einfügen läßt. Es wandelt alle positiven Zahlen zwischen 1 und 999 999 999 sowie die Null in Klartext um. Sie übergeben ihm in der Variablen <zahl> den Wert zur Umwandlung. Die Routine ab Zeile 2230 dient nur der Demonstration und gibt die gewandelte Zahl auf dem Bildschirm aus. Dort übernehmen Sie dann das Ergebnis für die weitere Bearbeitung.

(Valentin Gerber/ja)

| Steckbrief | |
|---|---|
| Programm: | Ziffwort |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

```
1000 REM ***************************      [F0FA]
1010 REM *   (c) Gerber Valentin    *      [DA4C]
1020 REM *       Dianastr. 16       *      [52CC]
1030 REM *       8013 Haar          *      [9588]
1040 REM ***************************      [0902]
1050 DIM s$(20),stelle(9)                  [4FD2]
1060 MODE 2                                [29BC]
1070 FOR k=0 TO 20:s$(k)="":NEXT           [CF26]
1080 LOCATE 7,10                           [080A]
1090 INPUT "Bitte geben Sie die umzuwand
     elnde Zahl in Ziffern ein: ",zahl     [8CB0]
```

**Listing. »Ziffwort« arbeitet als Demonstration auch allein, ist aber als Unterroutine in Programme einzubinden**

```
1100 CLS                                          [0188]
1110 IF zahl=0 THEN s$(1)="Null":GOTO 22
     10                                           [6254]
1120 IF zahl>10^9 THEN CLS:LOCATE 4,3:PR
     INT "Die eingegebene Zahl ist zu gr
     oss. Druecken Sie bitte irgend eine
      Taste.":CALL &BB18: GOTO 1060              [28B2]
1130 IF zahl<1 THEN CLS:LOCATE 4,3:PRINT
      "Die eingegebene Zahl ist zu klein
     . Druecken Sie bitte irgend eine Ta
     ste.":CALL &BB18: GOTO 1060                 [58EC]
1140 REM ********************************
     *****************                           [A7E8]
1150 REM Zerlegung der eingegebenen Zahl
      in ihre Ziffern                            [02AE]
1160 REM ********************************
     *****************                           [39EC]
1170 zahl$=STR$(zahl)                            [A02C]
1180 FOR ii=0 TO 9:stelle(ii)=0:NEXT             [ED30]
1190 FOR ziffer=0 TO LEN(zahl$)-1                [CE52]
1200      stelle(ziffer)=VAL(MID$(zahl$,L
     EN(zahl$)-ziffer,1))                        [092A]
1210 NEXT                                        [ED46]
1220 REM ********************************
     *****************                           [6DE6]
1230 REM           Bearbeitung der Million
     en                                          [8BE8]
1240 REM ********************************
     *****************                           [3FEA]
1250 i=1                                         [D77E]
1260 hunderter=stelle(8)                         [2612]
1270 zehner=stelle(7)                            [4288]
1280 einer=stelle(6)                             [1496]
1290 IF (stelle(8)+stelle(7)+stelle(6))=
     0 THEN GOTO 1370                            [B9EA]
1300 IF (stelle(8)+stelle(7))=0 AND stel
     le(6)=1 THEN s$(i)="eine":s$(i+1)="
     million":i=i+2:GOTO 1370                    [14E4]
1310 GOSUB 1530                                  [599C]
1320 s$(i+1)="millionen"                         [F0B6]
1330 i=i+2                                       [AAA6]
1340 REM ********************************
     *****************                           [3BEC]
1350 REM            Bearbeitung der Tausen
     der                                         [4D36]
1360 REM ********************************
     *****************                           [A9F0]
1370 hunderter=stelle(5)                         [0410]
1380 zehner=stelle(4)                            [B086]
1390 einer=stelle(3)                             [7694]
1400 IF (stelle(5)+stelle(4)+stelle(3))=
     0 THEN GOTO 1480                            [00CE]
1410 IF (stelle(5)+stelle(4))=0 AND stel
     le(3)=1 THEN s$(i)="ein":GOTO 1430          [B802]
1420 GOSUB 1530                                  [58A0]
1430 s$(i+1)="tausend"                           [DD14]
1440 i=i+2                                       [9BAA]
1450 REM ********************************
     *****************                           [72F0]
1460 REM            Bearbeitung der Hunder
     ter                                         [B346]
1470 REM ********************************
     *****************                           [00F4]
1480 hunderter=stelle(2)                         [C20E]
1490 zehner=stelle(1)                            [3684]
1500 einer=stelle(0)                             [1680]
1510 GOSUB 1530                                  [57A0]
1520 GOTO 2150                                   [9E12]
1530 zwischenwert=hunderter                      [A248]
1540 IF zwischenwert=0 THEN GOTO 1610            [B486]
1550 GOSUB 1930                                  [57B0]
1560 s$(i+1)="hundert"                           [DF28]
1570 i=i+2                                       [9AB2]
1580 REM ********************************
     *****************                           [D3F8]
1590 REM  Umwandlung der Einer- und Zehn
     erstellen bis 19                            [7744]
1600 REM ********************************
     *****************                           [ABEA]
1610 k=10*zehner+einer                           [88CA]
1620 IF k=0 THEN GOTO 2110                       [0DF8]
1630 IF k=1 AND stelle(0)=1 THEN s$(i)="
     eins":GOTO 2100                             [D886]
1640 IF k=1 THEN s$(i)="ein":GOTO 2100           [618C]
1650 IF k=2 THEN s$(i)="zwei":GOTO 2100          [B196]
1660 IF k=3 THEN s$(i)="drei":GOTO 2100          [B864]
1670 IF k=4 THEN s$(i)="vier":GOTO 2100          [848C]
1680 IF k=5 THEN s$(i)="fuenf":GOTO 2100
                                                 [CE4C]
1690 IF k=6 THEN s$(i)="sechs":GOTO 2100
                                                 [4854]
1700 IF k=7 THEN s$(i)="sieben":GOTO 210
     0                                           [CA06]
1710 IF k=8 THEN s$(i)="acht":GOTO 2100          [E75E]
1720 IF k=9 THEN s$(i)="neun":GOTO 2100          [D78E]
1730 IF k=10 THEN s$(i)="zehn":GOTO 2100
                                                 [25DE]
1740 IF k=11 THEN s$(i)="elf":GOTO 2100          [AEE6]
1750 IF k=12 THEN s$(i)="zwoelf":GOTO 21
     00                                          [03AA]
1760 IF k=13 THEN s$(i)="dreizehn":GOTO
     2100                                        [E532]
1770 IF k=14 THEN s$(i)="vierzehn":GOTO
     2100                                        [685A]
1780 IF k=15 THEN s$(i)="fuenfzehn":GOTO
      2100                                       [B21A]
1790 IF k=16 THEN s$(i)="sechzehn":GOTO
     2100                                        [183C]
1800 IF k=17 THEN s$(i)="siebzehn":GOTO
     2100                                        [2F2E]
1810 IF k=18 THEN s$(i)="achtzehn":GOTO
     2100                                        [D42C]
1820 IF k=19 THEN s$(i)="neunzehn":GOTO
     2100                                        [815C]
1830 IF einer=0 THEN GOTO 2020                   [7B4E]
1840 zwischenwert=einer                          [79D4]
1850 GOSUB 1930                                  [CAB6]
1860 s$(i+1)="und"                               [F7C8]
1870 i=i+2                                       [D2B8]
1880 GOTO 2020                                   [771C]
1890 REM ********************************
     *****************                           [7700]
1900 REM        Umwandlung der Hunderters
     tellen                                      [1BE6]
1910 REM        und Zehnerstellen von 20
     bis 99                                      [76BC]
1920 REM ********************************
     *****************                           [1DF4]
1930 IF zwischenwert=1 THEN s$(i)="ein":
     RETURN                                      [999C]
1940 IF zwischenwert=2 THEN s$(i)="zwei"
     :RETURN                                     [0BA6]
1950 IF zwischenwert=3 THEN s$(i)="drei"
     :RETURN                                     [3074]
1960 IF zwischenwert=4 THEN s$(i)="vier"
     :RETURN                                     [B79C]
1970 IF zwischenwert=5 THEN s$(i)="fuenf
     ":RETURN                                    [A25C]
1980 IF zwischenwert=6 THEN s$(i)="sechs
     ":RETURN                                    [9164]
1990 IF zwischenwert=7 THEN s$(i)="siebe
     n":RETURN                                   [D928]
2000 IF zwischenwert=8 THEN s$(i)="acht"
     :RETURN                                     [FC5C]
2010 IF zwischenwert=9 THEN s$(i)="neun"
     :RETURN                                     [BB8C]
2020 IF zehner=2 THEN s$(i)="zwanzig":GO
     TO 2100                                     [FF5E]
2030 IF zehner=3 THEN s$(i)="dreissig":G
     OTO 2100                                    [8602]
2040 IF zehner=4 THEN s$(i)="vierzig":GO
     TO 2100                                     [5152]
2050 IF zehner=5 THEN s$(i)="fuenfzig":G
     OTO 2100                                    [4712]
2060 IF zehner=6 THEN s$(i)="sechzig":GO
     TO 2100                                     [0A34]
2070 IF zehner=7 THEN s$(i)="siebzig":GO
     TO 2100                                     [5938]
2080 IF zehner=8 THEN s$(i)="achtzig":GO
     TO 2100                                     [FF36]
2090 IF zehner=9 THEN s$(i)="neunzig":GO
     TO 2100                                     [8666]
2100 i=i+1                                       [CC9C]
2110 RETURN                                      [CE88]
2120 REM ********************************
     *****************                           [4892]
2130 REM        Umwandlung in Grossbuchst
     aben                                        [DB20]
2140 REM ********************************
     *****************                           [DC96]
2150 buchstabe1$=LEFT$(s$(1),1)                  [6482]
2160 grossbuchstabe$=CHR$(ASC(buchstabe1
     $)-32)                                      [F376]
2170 s$(1)=grossbuchstabe$+RIGHT$(s$(1),
     LEN(s$(1))-1)                               [B29A]
2180 REM ********************************
     *****************                           [049E]
2190 REM            Ausgabe des Resultate
     s                                           [4C20]
2200 REM ********************************
     *****************                           [7390]
2210 z1$=s$(1)+s$(2)+s$(3)+s$(4)+s$(5)+s
     $(6)+s$(7)+s$(8)+s$(9)+s$(10)+s$(11
     )                                           [786E]
2220 z2$=s$(12)+s$(13)+s$(14)+s$(15)+s$(
     16)+s$(17)+s$(18)+s$(19)+s$(20)             [2312]
2230 m$="Die Zahl"+STR$(zahl)+" lautet i
     n Buchstaben:"                              [BFE0]
2240 LOCATE (80-LEN(m$))/2,8                     [8F58]
2250 PRINT m$                                    [894E]
2260 LOCATE (80-LEN(z1$))/2,10                   [CA2A]
2270 PRINT z1$                                   [2ACE]
2280 LOCATE (80-LEN(z2$))/2,11                   [1C32]
2290 PRINT z2$                                   [49D4]
2300 LOCATE 22,24                                [3C66]
2310 PRINT "Bitte druecken Sie irgendein
     e Taste."                                   [3A14]
2320 CALL &BB18                                  [276C]
2330 GOTO 1060                                   [8410]
```

**Listing. »Ziffwort« (Schluß)**

# Ergänzen Sie jetzt Ihre HAPPY-COMPUTER-Sammlung

## Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle »Happy Computer«-Ausgaben von 1985? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?
**Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der noch lieferbaren Ausgaben. Und so kommen Sie schnell an die gewünschten Ausgaben:** Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

| Stichwort | Titel | Seite/Ausgabe |
|---|---|---|
| | **Aktuelles** | |
| Computer | Amiga — ein Traumcomputer wird Wirklichkeit | 9/10 |
| | Atari: Lage gefestigt | 14/11 |
| | Der »Plus/4« ist endlich da | 12/2 |
| | Grundstein einer neuen Linie und kein zweiter PC | 13/10 |
| | Konsequentes Chaos (Der deutsche QL) | 14/10 |
| DFÜ | Akustikkoppler für C 64 | 9/1 |
| | Ascom-Koppler jetzt auch für Atari | 20/8 |
| | Ein Anschluß unter dieser Nummer (Mailbox Nummern) | 159/3 |
| | Mailboxbetrieb in den USA | 22/10 |
| | Neues DFÜ-Programm für den Spectrum | 22/10 |
| | Nullmodem zum Aufstecken | 12/1 |
| Software | Atari-Schreiber jetzt für 520 ST | 14/12 |
| | Software fast zum Nulltarif | 10/1 |
| | Träume werden wahr (Schneider-Neuheiten aus England) | 9/12 |
| Drucker | Mac Inker, der sparsame Drucker | 12/12 |
| Floppy | Commodore-Floppy auf Trab gebracht | 9/1 |
| | Diskettenlaufwerk für den Sharp MZ-800 | 12/1 |
| | Quick Disk — Die Floppy-Alternative (MSX) | 20/4 |
| Erweiterung | Mini-Expansion-Box für TI 99/4A | 11/1 |
| MSX | Das Musikwunder (Yamaha CX-5) | 141/2 |
| | Der Billig-MSX von Philips kommt | 50/1 |
| | CP/M mit MSX-Computer: so geht's | 141/2 |
| | Ein komplettes System von Philips | 19/8 |
| | Flotter Dreier (Sanyo, Goldstar und Canon) | 23/5 |
| | MSX-Mix | 45/3 |
| | Mit dem fliegenden Teppich auf Erfolgskurs | 15/10 |
| Bücher | Bücher zur DFÜ | 111/3 |
| | Bücher zum Denken (KI) | 120/10 |
| | **Messeberichte** | |
| | Die neuesten Heimcomputer (Winter-CES) | 9/3 |
| | Funkausstellung in Berlin: MSX war Trumpf | 9/11 |
| | Kampf der Kolosse (Winter-CES — Teil 1) | 9/4 |
| | Sommer-CES 1985: Weiche Welle in Chicago — Teil 1 | 9/8 |
| | Software-Jackpot (Winter CES — Teil 2) | 9/5 |
| | Software-Super-Show in London (PCW-Show) | 12/11 |
| KI | Künstliche Intelligenz in Wiesbaden (AI Europa) | 13/12 |
| Musik | Musikmesse Frankfurt: Midi marschiert | 22/5 |
| | **Interviews** | |
| | David Crane (Ghostbusters Autor) | 17/5 |
| | Interview mit den »Print Shop«-Machern | 14/8 |
| | Jack Tramiel (Chairman Atari) | 11/2 |
| | **Hardware-Tests** | |
| Drucker | Bewußt robust (Europrint K 6311 FT) | 31/5 |
| | **Software-Tests** | |
| Textverarb. | Ein Textprogramm, das sich lohnt (Homeword/C 64) | 77/4 |
| | Drei Drucker im Test (STX 80, Gemini 10X, CP-80X) | 16/1 |
| | (Nachhall auf Seite 149 in 4/85) DWX 305: Schönschrift zum Niedrigpreis | 18/2 |
| | Eine heiße Verbindung (EP 22, EP 44, EXD 10) | 26/5 |
| | Kompakt und leise: Matrixdrucker GLP (Centronics) | 24/1 |
| | Regenbogenfarben — wie gedruckt (Okimate 20) | 154/10 |
| | Schön oder schnell (Horizon HX 80) | 21/3 |
| | Spectrum mit starken Typen (Gabriele 9009) | 126/11 |
| | Zwei Drucker für den Schneider (NLQ 401, GP 500 CPC) | 112/8 |
| Computer | Chinese mit britischem Paß (Triton 64) | 22/2 |
| | Der Musik Maestro (Yamaha CX-5) | 28/4 |
| | Der Neue: Commodore PC 128 | 46/5 |
| | Der »neue« Spectrum | 31/1 |
| | Ein »Einsteiger« aus Taiwan (BIT-90) | 16/2 |
| | Joyce — Schneiders Einstieg in die Welt der PCs | 24/11 |
| | Quantensprung im Schneckentempo (QL dt. Version) | 180/11 |
| | Koreaner mit Deutsch-Talent (Ce-Tec/MSX) | 18/3 |
| | Schneiders neue Dimension (CPC 6128) | 24/10 |
| | Sharps Jüngster (Sharp MZ-800) | 20/1 |
| | Spectrum plus oder Spectrum minus | 24/4 |
| | Viel Computer für wenig Geld (Schneider CPC 664) | 113/8 |
| | YC-64: Fernöstlicher Biedermann (MSX Computer) | 20/2 |
| | Wer ist wer? (Atari 520 ST+ und 260 ST) | 16/12 |
| | Wie musikalisch ist mein Heimcomputer? | 148/11 |
| Laufwerke | 3-Zoll-Erfahrungen (MCD-1-Floppy für Spectrum) (Discovery/Spectrum) | 22/1 |
| | Ein ungleiches Paar (Spectrum — VIC 1541 Interface) | 21/4 |
| | Lauf, Floppy, lauf! (SpeedDos plus/C64) | 45/12 |
| | Preiswertes Spectr um Floppysystem (Viscount System) | 21/2 |
| | Spectrum Diskettensystem im Plus-Look | 20/3 |
| | VC 1541 wird zur Rennfloppy | 42/4 |
| Recorder | Der Spectrum Sprinter (Datenrecorder: Sprint) | 28/1 |
| | Ein billiger Speicher für alle (Recorder MC 3810) | 30/5 |
| DFÜ | DFÜ auch mit dem TI (RS 232 für TI 99/4A) | 25/5 |
| | Kommunikation mit dem Spectrum | 32/4 |
| | Spartanisch aber gut (Ascom Akustikkoppler) | 158/3 |
| Sonstiges | Computer steuert Modelleisenbahn | 176/11 |
| | Der andere Weg (Spectrum Tastatur) | 19/3 |
| | Faszination der Technik (Fischer Technik Roboter) | 44/11 |
| | Famose Formel für den C64 (Formel 64) | 40/12 |
| | Grafpad Supergrafik für den Spectrum | 16/3 |
| | Haltet den Dieb (Alarmanlage für C 64, VC 20) | 29/1 |
| | Ohren oder Tasten? (Voice Command Modul/C 64) | 40/10 |
| | Peripherie für MSX (Plotter, 3½-Zoll-Floppy) (Joysticks im Vergleichstest) | 26/1 |
| | Roboter, Technologie der Zukunft (Fischertechnik) | 45/4 |
| | Starker Arm für Heimcomputer (Teach Robot) | 38/4 |
| | Tafelfreuden für Grafik-Gourmets (Atari Maltafel) | 14/1/ |
| | Vom Piepmatz zum Mini-Orchester (Spectrum Sound) | 15/2 |
| | **Software-Tests** | |
| Textverarb. | Ein Textprogramm, das sich lohnt (Homeword/C 64) | 77/4 |
| | Jedem seine Zeitung (The Newsroom) | 118/8 |
| | Schreiben ohne Frust | 46/2 |
| | Textverarbeitung für jedermann (Homewriter für MSX) | 137/1 |
| Sprachen | Basic-Erweiterung zum Spartarif (Aztec Basic/C 64) | 76/4 |
| | Drei Assembler für Atari-Computer im Vergleich | 30/3 |
| | Fortschritt rückwärts (CP/M-80 Emulator für 520 ST) | 138/11 |
| | Hisoft-Pascal jetzt Microdrive-kompatibel (Spectrum) | 56/2 |
| | Logo für den Atari 520 ST | 134/11 |
| | Mallard-80-Basic — ein starkes Stück | 28/11 |
| | Maschinensprache ist keine Zauberei (CPC 464) | 107/8 |
| | Prozessor-Welt von morgen: C 64 simuliert 68000 | 42/10 |
| | Spezielles Spiele-Basic für den Spectrum | 143/5 |
| | Welches Basic für meinen MZ-700? | 48/2 |
| | Zwölf Farben in Mode 2 (Color Star für CPC 464) | 110/8 |
| Utilities | Das Programm, das Programme macht (Progressor) | 33/5 |
| | Disketten-Doktor für den C 128 | 42/12 |
| | Quicksave für Spectrum | 137/4 |
| | SM-Kit — Das Werkzeug für Lehrling und Meister (C 64) | 138/1 |
| | Software-Knackern dazwischengepfuscht (Apple II) | 27/3 |
| Grafik | Beeindruckend (Print Shop — Druckprogramm) | 50/2 |
| | Die Maus bringt Farbe auf den Bildschirm (Apple) | 52/2 |
| | Grafik gratis (Malprogramm Blazing Paddles) | 38/3 |
| | Koala Bilder zum Anfassen (Hardcopy-Programm) | 57/2 |
| | Mit dem Joystick programmiert (Designers Pencil) | 140/5 |
| | Viel Grafik für wenig Geld (Graphics Basic und Supergrafik 64 für C 64 im Vergleich) | 44/2 |
| | Vorsicht Kamera! (Take 1, Trickfilm Designer) | 126/8 |
| DFÜ | Apple II sucht Anschluß | 154/3 |
| | Contact 64 — Die Software zum Ascom-Koppler | 142/5 |
| | Spectrum auf Draht (DFÜ Vergleichstest) | 124/8 |
| Astronomie | Spectrums Sternstunden | 34/3 |
| | Sterngucker | 158/10 |
| Schach | Schachmatt per Telefon | 156/10 |
| | **Spiele-Tests** | |
| | Amazon | 145/5 |
| | Archon II: Adept | 126/2 |
| | Asylum | 144/3 |
| | Athletic Land | 146/1 |
| | A View to a Kill | 169/10 |
| | Ballblazer | 167/10 |
| | Boulder Dash | 125/2 |
| | Bounty Bob strikes back | 139/8 |
| | Cavelord | 124/2 |
| | Crazy Train | 144/1 |
| | D-Bug | 118/2 |
| | Deus ex Machina | 146/4 |
| | Don't buy this | 168/12 |
| | Doomdark's Revenche | 148/5 |
| | Dorodon | 142/3 |
| | Dragonsden | 124/2 |
| | Elektro Freddy | 145/1 |
| | Elite | 164/10 |
| | Eureka | 144/4 |
| | Fahrenheit 451 | 145/5 |
| | Five-a-Side Football | 166/10 |
| | Formula One | 140/8 |
| | Frank Brunos Boxing | 166/10 |
| | Frankie goes to Hollywood | 162/10 |
| | Fruity Frank | 145/4 |
| | Ghettoblaster | 169/11 |
| | Ghostbusters | 138/3 |
| | Ghost Chaser | 170/11 |
| | Great American Cross Country Road Race | 168/11 |
| | Hacker | 167/12 |
| | H.E.R.O. | 149/5 |
| | Hyper Sports 1 | 143/3 |
| | Karateka | 146/4 |
| | Kennedy Approach | 168/12 |
| | Knight Lore | 143/3 |
| | Macbeth | 144/4 |
| | Mask of the Sun | 122/2 |
| | Match Day | 150/5 |
| | Mindshadow | 141/8 |
| | Mr. Do | 167/10 |
| | Monster Trivia | 168/10 |
| | Nick Faldo plays the Open | 169/11 |
| | Nightshade | 169/12 |
| | Nodes of Yesod | 169/12 |
| | On Court Tennis | 150/5 |
| | Pitfall II | 148/5 |
| | Rama | 146/5 |
| | Rescue on Fractalus | 168/10 |
| | Rocket Ball | 140/8 |
| | Rockford's Riot (Bolder Dash II) | 168/11 |
| | Rock'n Bolt | 139/8 |
| | Sherlock Homes | 121/2 |
| | Seastalker | 147/1 |
| | Serpent's Star | 142/4 |
| | Software Star | 165/11 |
| | Spelunker | 142/3 |
| | Standing Stones | 145/4 |
| | Summer Games II | 133/8 |
| | Super Pipeline II | 141/8 |
| | The Ancient Art of War | 149/5 |
| | The Fourth Protocol | 165/11 |
| | The Hitchhiker's Guide to the Galaxy | 138/4 |
| | The Little Computer People Projekt | 170/12 |
| | The Way of exploding Fist | 169/10 |
| | Tour de France | 170/11 |
| | Where in the World is Carmen San Diego | 163/11 |
| | Whistler's Brother | 141/3 |
| | White Lightning | 148/1 |
| | Winter Games | 164/12 |
| | World Championship Boxing | 170/12 |
| | ZimSalaBim | 141/3 |
| | **Spiele Tips** | |
| | Abenteuer im Weltraum | 152/5 |
| | Amazon | 172/10 |
| | Aztec Challenge | 147/4 |
| | Aztec Tomb | 147/4 |
| | Aztec Tomb | 173/10 |
| | Beach Head | 85/1 |
| | Death in the Caribbean | 142/8 |
| | Death in the Caribbean | 172/12 |
| | Doomdark's Revenge | 142/8 |
| | Everyone's a Wally | 173/10 |
| | Ghostbusters | 140/3 |
| | Ghostbusters | 147/4 |
| | Ghostbusters | 152/5 |
| | Hampstead | 172/12 |
| | Heros of Karn | 173/12 |
| | Hexenküche | 173/12 |
| | Hobbit | 146/3 |
| | Hulk | 143/8 |
| | Hunch Back | 85/1 |
| | Karateka | 172/12 |
| | Lode Runner | 174/11 |
| | Lode Runner | 174/12 |
| | Mask of the Sun | 173/11 |
| | Masquerade | 144/8 |
| | Mindshadow | 174/11 |
| | Miner 2049er | 147/4 |
| | Pirate Adventure | 126/2 |
| | Pitfall | 147/4 |
| | Pitfall II | 144/8 |
| | Pitfall II | 174/10 |
| | Sabre Wulf | 85/1 |
| | Sands of Egypt | 173/11 |
| | Sands of Egypt | 174/12 |
| | Schloß des Grauens | 152/5 |
| | Secret Mission | 173/11 |
| | Ship of Doom | 172/12 |
| | Spelunker | 144/8 |
| | Strip Poker | 152/5 |
| | Summer Games | 144/8 |
| | Summer Games | 172/10 |
| | Super Huey | 171/10 |
| | The Dallas Quest | 174/10 |
| | The Institute | 172/10 |
| | The Institute | 173/12 |
| | The Quest | 173/11 |
| | The Witness | 146/3 |
| | Time Maschine | 173/11 |
| | Ultima II | 126/2 |
| | Ultima II | 151/5 |
| | Ultima III | 151/5 |
| | Valhalla | 173/11 |
| | Whistler's Brother | 144/8 |
| | Zeppelin | 152/5 |
| | ZimSalaBim | 142/8 |
| | Zork | 172/10 |
| | **Listings** | |
| Anwendung | Alle Neune (Jahresauswertung-Kegeln/C 64) | 67/5 |
| | (Adreßverwaltung/C 64) Besseres Basic ganz einfach Software Basic 3.0/C 64) | 67/3 |
| | Datenbank mit freiem Zugriff (C 64) | 86/1 |
| | Der Halleysche Komet kommt (MSX) | 76/10 |
| | Die Mini-Textverarbeitung (Spectrum) | 74/8 |
| | Nachhall auf Seite 160 in 9/85 | |
| | Do-it-yourself-Datenverwaltung (Mainfile II/C 64) | 53/3 |
| | Einblick ins Innenleben (Disassembler/CPC 464) | 86/5 |
| | Eine tolle Textverarbeitung für den Schneider (464) | 90/3 |
| | Nachhall auf Seite 85 in 5/85 | |
| | Funktionen optisch aufbereitet (VZ-200/Laser) | 95/3 |
| | Geregelte Finanzen mit dem Commodore 64 | 80/11 |
| | Nachhall auf Seite 117 in 8/85 | 87/4 |
| | Morse-Decoder für Funkamateure (Spectrum) | 117/11 |
| | Nebenkostenabrechnung (C 64) | 63/10 |
| | Nachhall auf Seite 80 in 12/85 Optik mit Simons Basic (C 64) | 63/3 |
| | Programme in Reih' und Glied (C 64) | 62/12 |
| | Suchen, nein danke (Dateiverwaltung/CPC 464) | 83/4 |
| | Nachhall auf Seite 117 in 8/85 | |
| | Transistor-Schaltungen berechnen (L.d.M./Spectrum) | 51/1 |
| | Turbo-Basic-Interpreter für Atari 800XL (L.d.M.) | 81/12 |
| Grafik | Apple IIc-HiRes-Grafik auf dem Drucker | 106/11 |
| | Bewegte Grafik mit drei Befehlen (CPC 464) | 74/10 |
| | Farbspielereien (Atari) | 89/3 |
| | Grafikentzerrung für Matrixdrucker (Spectrum) | 108/12 |
| | Grafik-Window bekommt Nachwuchs (C 64) | 68/10 |
| | Nachhall auf Seite 80 in 12/85 | |
| | Grafikzauber (Apple II) | 78/2 |
| | Hires Fantasy (C 64) | 58/3 |
| | Rosetten-Grafik für den Spectrum | 88/3 |
| | Schnelle Grafik aus dem Compiler (L.d.M./C 64) | 49/8 |
| | Schöne schnelle Grafik (Grafik-Paket/C 64) | 80/2 |
| | Solar-Painter (Spectrum) | 101/12 |
| | Sprite-Editor (C 64) | 82/1 |
| | Zauber der Farben mit Magic Painter (L.d.M./Atari) | 83/3 |
| | Nachhall auf Seite 86 in 5/85 | |
| | Zaubereien auf dem Bildschirm (L.d.M./Grafik/CPC 464) | 80/4 |
| | Zeichenroutine für Kreise und Ellipse (CPC 464) | 90/5 |
| | Zykloide für Grafiker und Mathematiker (C 64) | 60/10 |
| | Nachhall auf Seite 79 in 12/85 | |
| Spiel | Das Haus des Magiers (C 64) | 63/4 |
| | Dasher, der Volltreffer (L.d.M./C 64) | 62/5 |
| | Nachhall auf Seite 117 in 8/85 | |
| | Der rasende Raider (C 64) | 79/3 |
| | Diamantenfieber (L.d.M./Atari 48 KByte) | 58/2 |
| | Nachhall auf Seite 85 in 5/85 | |
| | Die Abenteuer eines rasenden Reporters (Report/C 64) | 60/1 |
| | Geröllheimer (Atari) | 79/5 |
| | Gespensterjagd im Schneider (CPC 464) | 74/2 |
| | Nachhall auf Seite 85 in 5/85 | |
| | Lumberjack Larrys Abenteuer in Bagdad (S.d.M/C 64) | 52/12 |
| | Kalte Zeiten (Wintry Screen/C 64) | 69/2 |
| | Kneipe zum hastigen Kellner (VC 20) | 68/1 |
| | Mit dem Apple auf die Trainerbank (Aktion Apfelsaft) | 100/4 |
| | Mit dem Atari-Computer auf Ölsuche (Atari) | 68/8 |
| | Mücke mit Tücke (C 64) | 70/1 |
| | Nachtflug (Spectrum) | 72/1 |
| | Nachhall auf Seite 85 in 5/85 Niemandsland (C 64) | 72/3 |
| | Pokerface für 16 KByte (Spectrum) | 76/5 |
| | Psycho — die Macht des Geistes (C 64) | 64/8 |
| | Nachhall auf Seite 80 in 12/85 | |
| | Rennfahrer mit dem Joystick (Driver/C 64) | 71/4 |
| | Rettet den letzten Baum (Insekt defense/C 64) | 72/2 |
| | SAM — der Mann von der Baustelle (L.d.M./CPC 464) | 109/11 |
| | Nachhall auf Seite 79 in 12/85 | |
| | Schatzhöhle (Atari 800XL) | 75/1 |
| | Vorsicht Hochwasser (Aquantor/L.d.M./C 64) | 54/10 |
| | Über den Wolken (Flugplanung/C 64) | 73/11 |
| | Wortsuchspiel (Spectrum) | 104/3 |
| Tips&Tricks | AMPEL — grünes Licht für Atari-Maschinen-Programme | 104/12 |
| | Auf Trap gebracht (CPC 464) | 73/12 |
| | Basic bequem (C 64) | 85/2 |
| | Basic-Compactor (Spectrum) | 82/10 |
| | Nachhall auf Seite 80 in 12/85 | |
| | Basic-Plus: Applesoft-Basic-Erweiterung (Apple II) | 96/3 |
| | Beim C 64 piepst es | 70/11 |
| | Bilder richtig konservieren (CPC 464) | 97/4 |
| | Bildschirmtrick für den Commodore 64 | 70/11 |
| | Byte-Shifter (Spectrum) | 91/5 |
| | Dateien hin- und hergerissen (Atari-IBM) | 94/4 |
| | Der neue Checksummer ist da (C 64) | 64/10 |
| | Der neue Checksummer ist da (C 64) | 69/11 |
| | Der neue Checksummer (C 64) | 67/12 |
| | Deutsche Sonderzeichen unter CP/M (CPC 464) | 69/12 |
| | Die Maltafel wird zur Maus (Atari) | 98/4 |
| | Nachhall auf Seite 49 in 11/85 | |
| | Disk-Help für die schnelle Hilfe (Atari) | 71/8 |
| | Disk- und DOS-Utility für alle Atari-Computer | 77/10 |
| | Drei Tricks für MSX | 92/5 |
| | Ein langes Gesicht für den C 64 (Longscreen 64) | 72/11 |
| | Fehlerhilfe mit HELP & TRACE (VC 20) | 95/1 |
| | Fensterkünstler (C 64) | 71/5 |
| | Fettschrift für den 48 KByte-Spectrum | 98/5 |
| | Find Label (Spectrum) | 88/2 |
| | Flotte Primzahlen in Hisoft Pascal (Spectrum) | 86/2 |
| | Fußball-Manager für Commodore 64 | 58/8 |
| | Grafik-Hardcopy in vierfacher Größe (C 64) | 61/8 |
| | Kostenlose Speichererweiterung (C 64) | 67/10 |
| | Listen leicht gemacht (C 64) | 94/1 |
| | Make DATA für den Spectrum | 102/3 |
| | Maschinencode-Routinen in Basic umgesetzt (CPC 464) | 75/10 |
| | Microdrive-Aufwertung (Spectrum) | 122/11 |
| | Nachhall auf Seite 79 in 12/85 | |
| | Mondlandung (C 64) | 55/8 |
| | Musik und Farbe (C 64) | 68/12 |
| | Nie mehr Listingkummer mit dem Checksummer (C 64) | 84/2 |
| | Nie mehr Listingkummer mit dem Checksummer (C 64) | 61/4 |
| | Prima Werkzeug für den Programmierer (C 64) | 69/8 |
| | Programmtransfer leicht gemacht (CPC 464) | 72/10 |
| | Proportionalschrift für den Spectrum | 94/5 |
| | Protokoll auf dem Drucker (CPC 464) | 76/8 |
| | RAM-Disk für Atari 800XL | 119/11 |
| | Ran an den Userport (C 64) | 72/11 |
| | Renumber 64 (C 64) | 70/5 |
| | Nachhall auf Seite 117 in 8/85 | |
| | Rock me Amadeus (C 64) | 66/11 |
| | RSX-Befehle ohne »@« (Schneider) | 73/12 |
| | Schilderwald (Plakatschrift/C 64) | 66/10 |
| | Schluß mit der Eintönigkeit (C 64) | 70/11 |
| | Nachhall auf Seite 80 in 12/85 | |
| | Spectrums COPY besser nutzen | 83/10 |
| | Spectrumtasten mit Funktionen belegt (Spectrum) | 98/1 |
| | Spectrum Tips & Tricks | 106/12 |
| | Sprachkurs für Commodore-Basic (C 64) | 90/1 |
| | Super-Merge für Commodore 64 | 54/8 |
| | Nachhall auf Seite 160 in 9/85 | |
| | Super-Saver (C 64) | 67/10 |
| | Statuszeile mit Uhr (Atari) | 96/1 |
| | Tasword 464 mit DIN-Tastatur (CPC 464) | 73/10 |
| | Texte auch im Grafikmodus (Atari) | 92/5 |
| | Tippen mit dem Plotter (C 64) | 94/1 |
| | Tips & Tricks rund um den Schneider | 77/8 |
| | Töne aus dem Atari | 96/5 |
| | Variablendump für Atari (Atari) | 83/2 |
| | Variablen-Transfer (Spectrum) | 123/11 |
| | Verflixter Listschutz (C 64) | 68/10 |
| | Nachhall auf Seite 80 in 12/85 | |
| | Vom Maschinencode zum Basic-Programm (C 64) | 66/10 |
| | Wie die Bilder laufen lernten (Atari) | 86/2 |
| | Zeilenakrobatik auf dem Schneider | 78/8 |
| | Zwei SCREEN$ im schnellen Wechsel (Spectrum) | 73/8 |
| | ZX81-Utility: Nützliches für Aufsteiger (C 64) | 61/11 |
| | 30 tolle Maschinencode-Routinen (Spectrum) | 98/3 |
| | **Grundlagen** | |
| Speicher | Daten am laufenden Band | 26/8 |
| | Daten auf der schnellen Scheibe | 30/8 |
| | Floppy gegen Kassette | 38/8 |
| | Selbst geschraubt ist halb gespart | 39/8 |
| | So arbeitet das 1050-Laufwerk von Atari | 36/8 |
| | So liest und schreibt die 1541 | 34/8 |
| | Speichermedium Endlosband | 28/8 |
| | Tips, Tricks und Todsünden | 41/8 |
| | Wohin in Zukunft mit Bits und Bytes | 24/8 |
| Monitore | Farbmonitore — buntes Fenster zum Computer | 127/5 |
| | Monitore: Richtig geplant, gekauft und genossen | 133/5 |
| Drucker | Blitzsaubere Schrift mit Laserlicht (Laserdrucker) | 147/10 |
| | Die »heißen« Drucker (Thermodrucker) | 139/10 |
| | Farbspiele für Farbdrucker | 135/10 |
| | Mit leisen Tönen (Tintenstrahldrucker) | 128/10 |
| | Scharfe Nadel, spitze Typen (Matrix und Typenrad) | 145/10 |

**Die Ausgaben 6/85, 7/85 und 9/85 sind bereits vergriffen und nicht mehr lieferbar!**

## Auch die bisher erschienenen Sonderhefte können Sie jetzt direkt bestellen:

**SONDERHEFT 01/84: SINCLAIR**
*Unentbehrliche Informationen zu den Sinclair Computern ZX81 und Spectrum.*

**SONDERHEFT 01/85: SPECTRUM**
*Anwendungsbezogene Listings und Tips & Tricks für alle Spectrum-Fans.*

**SONDERHEFT 02/85: SCHNEIDER 1**
*Eine Fülle wertvoller Beiträge und Listings für alle Schneider-Anwender.*

**SONDERHEFT 03/85: SPIELE**
*Ein Super-Nachschlagewerk für alle Spiele-Fans mit 100 Spielen im Test und großer Marktübersicht.*

**SONDERHEFT 01/86: SCHNEIDER 2**
*Noch mehr Tips und Tricks für Einsteiger und Fortgeschrittene mit vielen interessanten Programm-Listings.*

**SONDERHEFT 02/86: ATARI 1**
*Besonders 800 XL- und 130 XE-Fans erwarten jede Menge Anwendungs- und Spiele-Listings sowie Informationen.*

**SONDERHEFT 03/86: 68000er**
*Umfassende Informationen zur neuen Computer-Generation und eine große Vergleichstabelle, die im Detail über alle 68000er informiert.*

**SONDERHEFT 04/86: SCHNEIDER 3**
*Eine Erweiterung für alle Schneider-Anwender, Super-Programm-Listings und großer Einsteiger-Teil.*

**SONDERHEFT 05/86: PROGRAMMIERSPRACHEN**
*Fuß fassen in »Pascal«, »C« und »Forth« mit jeweils einem grundlegendem Kurs und vielen Anwendungs-Listings.*

**SONDERHEFT 06/86: 68000er 2**
*Umfangreicher Listingteil, viele Informationen, Tips und Tricks für Anwender der 68000er-Computer.*

**SONDERHEFT 07/86: SCHNEIDER 4**
*Mit den Schwerpunkten Joyce und CP/M plus, Ratschlägen zur Vortex-Karte und vielen Tips & Tricks.*

**SONDERHEFT 08: COMPUTER ALS HOBBY**
*Wissenswertes für Einsteiger und zusätzliche Informationen zur Fernsehsendung Computerzeit.*

**SONDERHEFT 09: 68000er 3**
*Mit den Schwerpunkten Sound- und Videodigitalisierung und Spieleprogrammierung.*

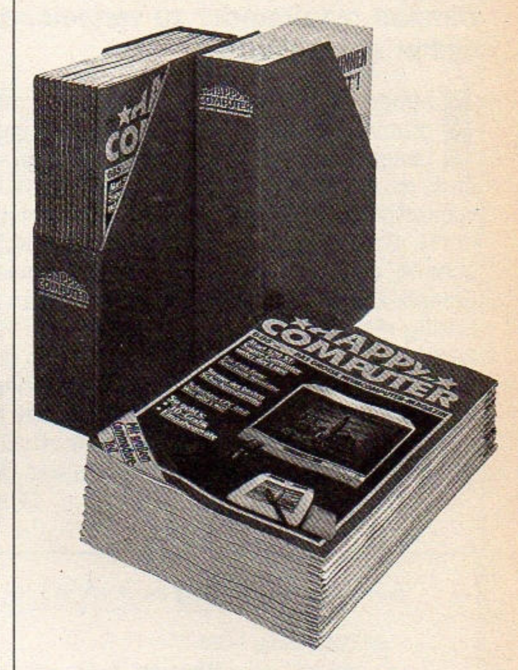
**SONDERHEFT 10/86: SCHNEIDER 5**
*Der neue Schneider-PC wird vorgestellt. Wieder viele Hilfestellungen und Kurse.*

**SONDERHEFT 11/86: SPIELE-TESTS**
*Alles über aktuelle Spieletests, Computerprogramme, Grafik- und Musik-Software.*

**SONDERHEFT 12/86: 68000er 4**
*Ausführliche Testreihe aller Grafikprogramme für Atari ST, Amiga und Sinclair QL sowie viele Grundlageninformationen zu diesen Computern.*

Tragen Sie die Nummer des gewünschten Sonderheftes (z.B. 08/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein.

## Am besten gleich mitbestellen: Die Happy-Computer-Sammelboxen




Für alle Leser, die »Happy Computer« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es ein interessantes Service-Angebot: die Happy-Computer-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk. Ein kompletter Jahrgang (12 Ausgaben) paßt in eine der praktischen Sammelboxen!

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

# Einerlei?

**Es gibt viele Wege, die Inhalte verschiedener Dateien miteinander zu vergleichen. »Fcomp« ist sicher der leichteste.**

Immer wieder wünscht man sich, man hätte ein Programm, das zwei Programme auf Diskette vergleicht und Unterschiede auf dem Bildschirm ausgibt. Das passiert stets dann, wenn man zum Beispiel ein neues Programm entwickelt oder verändert. »Welche Version ist denn nun die neuere?«. Solche und ähnliche Fragen beantwortet in Zukunft schnell und vor allen Dingen vollautomatisch unser Programm. Einzige Vorbedingung: Sie geben das Turbo-Pascal-Listing »Fcomp« ein und compilieren es. Rufen Sie dann die fertige Utility auf. Sie brauchen nur noch die Dateinamen der beiden Probanden einzugeben und können sich gemütlich zurücklehnen. Fcomp zeigt Unterschiede dann mit der Nummer des unterschiedlichen Bytes innerhalb der Datei an. Auf Wunsch erfolgt die Ausgabe auch über Drucker.

(Thomas Bullinger/ja)

```
            Filecomp
          Version 1.0

Drucker (j/n) ? n

1. Datei ? program
2. Datei ? program.bak

02EB: 35 30 , 023F: 32 24 , 0241: 31 43 , 0242: 35 , 48 ,
0243: 2B 52 , 0244: 49 24
```

Die Unterschiede zweier Dateien auf einen Blick

| Steckbrief | |
|---|---|
| Programm: | Fcomp |
| Computer: | CPC 464/664/6128/Joyce/PC |
| Datenträger: | Diskette |
| Besonderes: | Turbo-Pascal-Programm |

```
program FileComp; { Vergleich zweier Dateien } {U+}

const block_groesse  = 128;
      maxblock       = 512;  { max. 64 KByte pro Datei }
      maxver         = 1024; { max. 1 K Vergleiche }

var   filename       : string[14];
      file_1, file_2 : file;
      file_1_groesse : integer;
      file_2_groesse : integer;

      block_nr       : byte;
      block_index    : integer; { Adresse eines Unterschiedes }
      block_puffer_1 : array[1..512] of byte;
      block_puffer_2 : array[1..512] of byte;

      ver_index      : integer;
      ver_file       : text;

      ende_flag      : boolean;
      druck          : boolean;
      druck_char     : char;

procedure WriteHex (b:byte);

var   b1:byte;

procedure WriteNibble (b:byte);
begin
  b := b + $30;
  if (b > $39) then
  b := b + 7;
  write(chr(b));

  if (druck) then
    write(lst,chr(b));

end; { WriteNibble }

begin
  b1 := b shr 4;
  WriteNibble (b1);
  b1 := b and $0f;
  WriteNibble (b1);
end; { WriteHex }

procedure ver_block; { Vergleich zweier Bloecke }

var i : integer;

begin
  i := 1;
  while ((not ende_flag) and (i <= block_groesse)) do
  begin
    if (ver_index < maxver) then { nur bis zur Grenze des Erlaubten! }
    begin
      if (block_puffer_1[i] <> block_puffer_2[i]) then
      begin  { Unterschied gefunden }
        block_index := i + 255 + ((block_nr-1) * block_groesse);
        writehex(hi(block_index)); { Adresse anzeigen }
        writehex(lo(block_index));
        write(': ');
        if (druck) then
          write(lst,': ');
        writehex(block_puffer_1[i]); { 1. Datum }
        write(' ');
        if (druck) then
          write(lst,' ');
        writehex(block_puffer_2[i]); { 2. Datum }
        write(' , ');
        if (druck) then
          write(lst,' , ');
        if ((ver_index mod 4) = 0) then { Nach 4 Adressen }
        begin { Zeilenvorschub }
          writeln;
          if (druck) then
            writeln(lst);
        end;
        ver_index   := ver_index + 1;
      end;
    end
    else  { Mehr als max. Unterschiede gefunden }
    begin
      ver_index := maxver;
      ende_flag := true;
    end;
    i := i + 1;
  end; { Schleife }
end; { ver_block }

begin { Hauptprogramm }
  clrscr; { LOGON auf Bildschirm }
  highvideo;
  writeln('                              ');
  writeln('          Filecomp            ');
```

```
  writeln('       Version 1.0       ');
  writeln('                         ');
  lowvideo;
  repeat { Drucker ja oder nein }
    gotoxy(1,6);
    clreol;
    write ('Drucker (j/n) ? ');
    readln(druck_char);
  until (druck_char in ['j','J','n','N']);
  druck_char := upcase(druck_char);
  druck      := (druck_char = 'J');
  if (druck) then
    writeln(lst);
  repeat { Filename 1 erfragen }
    gotoxy(1,8);
    clreol;
    write('1. Datei ? ');
    readln(filename);
    assign(file_1,filename);
    {$I-} reset(file_1); {$I+}
  until (ioresult = 0);
  if (druck) then
    writeln(lst,'1. Datei : ',filename);
  repeat { Filename 2 erfragen }
    gotoxy(1,9);
    clreol;
    write('2. Datei ? ');
    readln(filename);
    assign(file_2,filename);
    {$I-} reset(file_2); {$I+}
  until (ioresult = 0);
  if (druck) then
  begin
    writeln(lst,'2. Datei : ',filename);
    writeln(lst);
  end;
  writeln;
  block_nr  := 1;
  ver_index := 1;
  file_1_groesse := filesize(file_1);
  file_2_groesse := filesize(file_2);
  if ((file_1_groesse > maxblock) or (file_2_groesse > maxblock)) then
    ende_flag := true
  else
    ende_flag := false;
  while (not ende_flag) do { Ueberpruefung bis zum Ende }
  begin
    blockread(file_1,block_puffer_1,1);
    blockread(file_2,block_puffer_2,1);
    ver_block;
    if (not ende_flag) then
    begin
      block_nr := block_nr + 1;
      if ((block_nr > file_1_groesse) or (block_nr > file_2_groesse)) then
        ende_flag := true
      else
        ende_flag := false;
    end;
  end;
  writeln;
  writeln('Es wurden ',ver_index - 1, ' Unterschiede gefunden.');
  writeln;
  if (druck) then
  begin
    writeln(lst);
    writeln(lst,'Es wurden ',ver_index, ' Unterschiede gefunden.');
    writeln(lst);
  end;
end.
```

**Listing. Wer kein Turbo Pascal besitzt, findet auf der Leserservice-Diskette die lauffähige COM-Datei des Programms**

# Turbo macht sich dünn

**Der Platz auf einer 3-Zoll-Diskette von Schneider ist rar. Verzichten Sie auf die Bibliothek von Turbo Pascal, und bei jedem Programm gewinnen Sie 8 KByte.**

Ein unter Turbo Pascal geschriebenes Programm läuft nur, wenn der Programmcode auch die Bibliothek enthält. In dieser Bibliothek stehen die Routinen, die den Computer zu seiner Arbeit anhalten.

Jedes Programm enthält diese Bibliothek in genau gleicher Form. Sie steht so natürlich auch mehrmals auf der Diskette. Hält man den Compiler nun dazu an, die Routinen nicht in den Code einzubinden, sondern einen Aufruf der in einer Sammeldatei stehenden Programmteile einzusetzen, dann spart man pro Programm 8 KByte Speicherplatz.

Programme ohne Bibliothek kennzeichnet unter Turbo Pascal der Zusatz ».CHN«, lauffähige Programme mit Bibliothek hingegen ein ».COM«. Das Programm »Command« im Listing wird mit der Vorgabe »C« compiliert, enthält also die Bibliothek. Alle anderen Programme, die auf der Diskette stehen, müssen mit »H« compiliert werden, enthalten also nur den eigentlichen Code. Starten Sie nun Command, so fragt der Computer Sie nach der aufzurufenden Datei. Diesen Namen geben Sie ohne den Zusatz ».CHN« an.

Die Eingabe wird so lange angefordert, bis ein gültiger Name genannt oder bis <CTRL-C> gedrückt wird. Um Komplikationen zu vermeiden, muß die Startadresse der Programme immer gleich sein. (Lothar Paucker/hg)

| Steckbrief | |
|---|---|
| Programm: | Command |
| Computer: | CPC 464/664/6128/Joyce/PC |
| Datenträger: | Diskette |
| Besonderes: | Turbo-Pascal-Programm |

```
program command;

var filename: file;
    com     : string(.8.);

begin
  repeat
    clrscr;
    write ('Command: ');
    readln (com);
    (*$I-*)
    assign (filename,com+'.CHN');
    chain (filename);
    (*$I+*)
  until ioresult = 0;
end.
```

**Listing. Platzsparen mit »Command«**

# »Bad sector« entschärft

**Die Fehlermeldung »Bad sector« unter CP/M 2.2 bereitet vielen Anwendern Kopfzerbrechen. Das muß nicht sein.**

Das Schlimmste, was einem bei CP/M 2.2 passieren kann, ist die Fehlermeldung »BDOS Error: Bad sector«. Das bedeutet nicht nur, daß die verwendete Diskette defekt ist, sondern auch, daß CP/M nach dem Drücken einer Taste das laufende Programm abbricht. So kann neben dem Ärgernis, daß ein Sektor der Diskette unleserlich ist, der weitaus größere Schaden auftreten, daß wertvolle Daten, die im unterbrochenen Programm bearbeitet wurden, verlorengehen, weil der Anwender sie nicht mehr speichern konnte.

CP/M 2.2 besitzt nun eine Besonderheit, die jedoch selten beschrieben wird: Wenn Sie nach der Fehlermeldung die ENTER-Taste drücken, ignoriert CP/M den Fehler. Dadurch erhalten Sie die Gelegenheit, die Daten im Speicher zu sichern und die Arbeit mit einer einwandfreien Diskette zu wiederholen.

Unter CP/M Plus tritt das Problem bei defekten Sektoren nicht auf, denn hier können Sie mit »Ignore« den defekten Sektor ausklammern. (Martin Kotulla/ma)

---

# GRAPHICS PEN auf CPC 464

**CPC 664 und 6128 kennen den Befehl GRAPHICS PEN, der die Farbe des Grafikstifts festlegt. Auf dem CPC 464 können Sie diesen Befehl simulieren.**

Jeder CPC-Besitzer, der häufig und gern Grafik auf seinem Gerät programmiert, möchte öfters die Farbe des Grafikstiftes, der über die X/Y-Koordinaten gesteuert wird, ändern. Auf dem CPC 664 und 6128 geschieht dies einfach mit dem Befehl GRAPHICS PEN. Leider fehlt dieser Befehl beim CPC 464. Besitzer dieses Computermodells können den Befehl jedoch durch folgende Befehlsfolge simulieren:

```
x=XPOS:y=YPOS:PLOT 800,800,penfarbe:MOVE x,y
```

Den Variablen x und y werden die aktuellen Grafik-Koordinaten zugewiesen. Der PLOT-Befehl setzt einen unsichtbaren Punkt (die Koordinaten liegen außerhalb des Bildschirms) und ändert die Stiftfarbe in den Wert der Variablen penfarbe. Anschließend wird der Grafikcursor wieder auf die alte Position gesetzt. (Martin Kotulla/ma)

---

# Reset auf Umwegen

**Ein versteckter RESET-Befehl löscht Ihre Programme, wenn ein Unbefugter darauf zugreifen möchte.**

Daß CALL 0 oder die Eingabe des RSX-Befehls IBASIC einen Reset auf dem Schneider CPC hervorruft, wissen die meisten CPC-Besitzer. Wer diese Befehle in seinem Programm an geeigneter Stelle versteckt, erreicht, daß das Programm abstürzt, sobald es gestartet wird. Nur der Anwender selbst weiß, wo der Befehl entfernt werden muß, damit das Programm einwandfrei funktioniert.

Dieses Verfahren ist jedoch mittlerweile so bekannt, daß der Programmschutz von jedem halbwegs informierten Programmierer entfernt werden kann. Neu ist dagegen folgende Lösung, die ihr Geheimnis nicht so schnell preisgibt:

```
10 POKE &BB5A,&C7
```

Dieser Befehl verbiegt den Vektor für die Bildschirmausgabe auf den Maschinensprache-Befehl RST 0, der einen Reset auslöst. Der erste PRINT-Befehl im Programm, der auf diesen POKE-Befehl folgt, löst einen Reset aus, der das Programm und alle Daten löscht.

Wenn Sie den Befehl gut in Ihrem Programm verstecken, die beiden hexadezimalen Werte nicht direkt, sondern über Variablen zuweisen und das Ganze mit GOSUBs verschachteln, ist die Chance ziemlich groß, daß dieser Schutz von fremden Programmierern nicht entdeckt wird. Sie müssen nur noch darauf achten, daß Ihr Programm auch einen PRINT-Befehl enthält, der den Reset auslöst. (Martin Kotulla/ma)

---

# SUBMIT ohne Bildschirmausgabe

**Viele Anwender empfinden es als störend, daß CP/M die Zeilen einer SUBMIT-Datei bei der Ausführung auf den Bildschirm ausgibt.**

Auf die Dauer geht es etwas an die Nerven, daß bei der Ausführung einer SUBMIT-Datei jedesmal die Befehlszeilen angezeigt werden. Um wieviel professioneller würde es wirken, wenn sich die Bildschirmausgabe unterdrücken ließe. Unter MS-DOS bewirkt der Befehl ECHO OFF das Abschalten der optischen Anzeige, doch unter CP/M gibt es offiziell keinen vergleichbaren Befehl.

Unter CP/M Plus läßt sich jedoch ein Trick anwenden, der die Bildschirmausgabe abschaltet. Das Standardprogramm DEVICE.COM gestattet es, die Zuweisung der logischen und physikalischen Peripheriegeräte zu verändern. Legen Sie doch einfach die Bildschirmausgabe (CONOUT) auf das Null-Device (NUL):

```
A>DEVICE CONOUT:=NUL
```

Sowohl im Direktmodus als auch in SUBMIT-Dateien werden von nun an alle Bildschirmausgaben unterdrückt. Am Ende eines SUBMIT-Durchlaufs wird die Bildschirmausgabe wieder eingeschaltet:

```
A>DEVICE CONOUT:=CRT
```

(Martin Kotulla/ma)

---

# DDT zeigt Grafikzeichen

**Nur ein kleiner Patch versetzt den CP/M-Debugger DDT.COM in die Lage, auch Grafikzeichen darzustellen.**

Der Dump-Befehl »D« des DDT-Debuggers listet einen Speicherauszug auf dem Bildschirm oder Drucker auf. Alle Datenbytes mit Werten zwischen 31 und 127 werden als ASCII-Zeichen dargestellt. Bytes, die nicht innerhalb dieses Wertebereichs liegen, symbolisiert der Debugger durch Punkte.

Dieses Verfahren ist für den Anwender nicht akzeptabel, da






der Schneider CPC die ASCII-Zeichen ab dem Wert 127 als Grafikzeichen auf dem Bildschirm ausgibt. Da hätte man natürlich auch gerne, daß der Debugger die Grafikzeichen beim Auflisten des Speicherinhalts anzeigt.

Dies ist mit einem kleinen Patch machbar. Lassen Sie den Debugger sich selbst aufrufen, und ändern Sie den Inhalt der Speicheradresse 0E37 hex in FF hex um. Dieser Wert gibt die Obergrenze der Datenbyte-Werte an, die der Debugger als ASCII-Zeichen darstellt. Anschließend speichern Sie den gepatchten DDT wieder auf Diskette.

So gehen Sie im einzelnen vor:

```
A>DDT DDT.COM
-SOE37
0E37 7F -> FF
0E38 ^C
A>SAVE 19 DDT.COM
```

(Martin Kotulla/ma)

# Zeichendefinition geht doch

**Konnten Sie Ihrem Schneider-Drucker DMP-2000 schon einmal selbstdefinierte Zeichen entlocken? Unser kleiner Tip sagt Ihnen, wie das geht.**

Jeder, der mit Hilfe des Beispielprogramms im Handbuch des DMP-2000 schon einmal versucht hat, seinem Drucker zu einem neuen Zeichensatz zu verhelfen, scheitert bald. Das liegt aber nicht am Programm, sondern am falsch eingestellten Dip-Schalter. Der vierte Schalter auf der zweiten Leiste (DS2-4) muß nämlich auf »ON« stehen. Wenn Sie den Drucker dann aus- und wieder eingeschaltet haben, läuft das Programm einwandfrei. (Michael Strasser/hg)

# TYPE mit Wildcards

**Der TYPE-Befehl hat einen großen Nachteil: Er erlaubt nur eindeutige Dateinamen. Eine Konstruktion wie TYPE *.TXT ist unmöglich. Auch PIP CON:=*.TXT funktioniert nicht.**

Gelegentlich möchte der Anwender sich sämtliche Textdateien einer Diskette anschauen. Der TYPE-Befehl erlaubt allerdings keine Eingabe von Wildcards, so daß man alle Dateinamen von Hand eingeben muß.

Ein kleiner Trick hilft, um die Eingaben aller Dateinamen herumzukommen. Kopieren Sie dazu mit PIP.COM die Dateien auf eine andere Diskette (am besten eine RAM-Disk) und geben Sie als Option »Echo« an:

```
A>PIP C:=A:*.TXT[E]
```

Hierdurch gibt CP/M beim Kopieren alle Dateien auf den Drucker aus. Wenn die Textausgabe beendet ist, können Sie selbstverständlich die kopierten Textdateien wieder löschen.

Mit <CTRL+P> erhalten Sie zusätzlich die Möglichkeit, den Befehl »PIP LST:=*.TXT« zu simulieren.

PIP hat bei der Echo-Funktion die unangenehme Eigenschaft, das EOF-Zeichen 26 mit auszudrucken, wodurch unter CP/M 2.2 ein Fenster gesetzt wird. Deshalb sollten Sie vor dem Ausdruck das Programm CTLOFF (Happy-Computer, Ausgabe 9/86, Seite 76) starten, das den Ausdruck des Steuerzeichens verhindert. Unter CP/M Plus ist diese Vorsichtsmaßnahme nicht nötig. (Martin Kotulla/ma)

# RAM-Disk ohne Gefahren

**Wer bei der Textverarbeitung statt Diskette eine RAM-Disk verwendet, bedenkt oft nicht, daß bei einem Absturz des Computers der gesamte Text verlorengeht.**

Die Textverarbeitung mit Wordstar bei Einsatz der Vortex-Speichererweiterung als RAM-Disk ist sehr komfortabel. Peinlich wird es, wenn der Strom ausfällt. Der gesamte Text ist gelöscht, weil Wordstar mit dem Kommando <CTRL+K> und <S> die Teil-Sicherungen des Textes nicht auf Diskette, sondern in die RAM-Disk gespeichert hat.

Spätestens beim ersten Computerabsturz wünscht sich der Anwender, daß sein Wordstar den Text gelegentlich auf Diskette sichert. Gesucht wird eine Befehlsfolge, die den Text ab und zu als Datei auf eine Diskette in Laufwerk A zwischenspeichert. Folgende Tastenkombinationen erfüllen diese Forderung:

<CTRL+K> und <9>,<br>
<CTRL+Q> und <R>,<br>
<CTRL+K> und <B>,<br>
<CTRL+Q> und <C>,<br>
<CTRL+K> und <K>,<br>
<CTRL+K> und <A>,<br>
A:WSDATEI,<br>
<CTRL+M> und <J>,<br>
<CTRL+Q> und <9>,<br>
<CTRL+K> und <9>,<br>
<CTRL+K> und <H>

Diese Tastenfolge müssen Sie nicht jedesmal eingeben, wenn Sie Ihren Text zwischenspeichern wollen. Patchen Sie stattdessen mit SETUP.COM (Amstrad-CP/M) oder INSTALL.COM (Vortex-CP/M) die Tastaturbelegungstabelle von CP/M so, daß die Zeichenkette auf einer der Funktionstasten liegt. Bei INSTALL.COM müssen Sie allerdings die Control-Codes in hexadezimale Werte umrechnen.

Und so arbeitet die Zeichenkette:

<CTRL+K> und <9>: Merker 9 an aktueller Position setzen<br>
<CTRL+Q> und <R>: An Textanfang springen<br>
<CTRL+K> und <B>: Blockanfang setzen<br>
<CTRL+Q> und <C>: An Textende springen<br>
<CTRL+K> und <K>: Blockende setzen<br>
<CTRL+K> und <W> mit Eingabe von A:WSDATEI: Block auf Laufwerk A als WSDATEI speichern<br>
<CTRL+M> und <J>: An Textanfang Leerzeile und »J« einfügen<br>
<CTRL+Q> und <9>: Alte Cursorposition anspringen<br>
<CTRL+K> und <9>: Merker 9 löschen<br>
<CTRL+K> und <H>: Block sichtbar machen

Wenn Sie das erste Mal die Taste drücken, der diese Tastenkombination zugewiesen wurde, müssen Sie nachher noch ein »J« vom Bildschirm löschen. Bei den nächsten Sicherungskopien ist das nicht mehr nötig, weil jetzt das »J« eine Sicherheitsabfrage von Wordstar beantwortet. Auf diese Weise sind Sie nicht jedesmal gezwungen, die Abfrage selbst zu beantworten.

Sobald Ihr Computer einmal abstürzt, können Sie aus der Datei WSDATEI den Text rekonstruieren – immer vorausgesetzt, Sie haben nicht vergessen, den Text regelmäßig zu sichern! Das beste Verhältnis zwischen Zeitverzögerung und Sicherheit erreichen Sie, wenn Sie alle 10 bis 15 Minuten abspeichern. (Martin Kotulla/ma)

# Start über Cursor-Tasten

**Disketten-Benutzer atmen auf. Der Aufruf von Programmen ist jetzt so einfach wie noch nie.**

Wer jemals mit dem CPC in Verbindung mit einem Kassettenrecorder gearbeitet hat, weiß die komfortable Art zu schätzen, wie dort Programme zu starten sind. Es genügt nämlich der gleichzeitige Druck zweier Tasten: <CTRL> und die kleine Taste <ENTER>. Nun bedarf es aber auch bei Diskettenbetrieb keines aufwendigen Programms, um ähnliche Voraussetzungen zu schaffen. Eine Zeile, genaugenommen nur ein einziger Befehl, ist alles:

```
KEY 140,STRING$(12,224)+CHR$(250)+"RUN"
+CHR$(34)+CHR$(13)
```

Wenn Sie dieses Kommando als Programm speichern, reicht der einmalige Aufruf, die neue Funktion zu installieren. Sie bleibt solange erhalten, bis Sie die Tasten neu belegen oder einen Reset ausführen. Woher aber »weiß« der Computer nun, welches Programm von der Diskette Sie laden und starten wollen? Ganz einfach. Geben Sie zuerst den Befehl CAT ein. Nachdem dann die Directory sichtbar ist, bewegen Sie den Cursor mit Hilfe der Cursor-Steuertasten auf den ersten Buchstaben des gewählten Dateinamens. Dort angelangt, genügt die genannte Tastenkombination, den Vorgang abzuschließen.

(Stefan Aust/ja)

```
10 '--> CAT+ -- (C)1986 von Stefan M.Aus
   t                                        [332A]
20 '                                        [6052]
50 KEY 140,STRING$(12,224)+CHR$(250)+"RU
   N"+CHR$(34)+CHR$(13)                     [A5FE]
```

**Listing. Programmstart per einfachem Tastendruck**

| Steckbrief | |
|---|---|
| Programm: | Cat Plus |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Diskette |

# AUTO neu aufgelegt

**Wer sich schon immer für seinen CPC 464 einen AUTO-Befehl wünschte, der so arbeitet wie derjenige der beiden neueren Modelle CPC 664 und 6128, ist mit dem Listing »AUTO Plus« fein heraus.**

Das Locomotive-Basic 1.1 der beiden Schneider Computer CPC 664 und CPC 6128 bietet gegenüber der Version 1.0 im CPC 464 einige Vorzüge. Zu diesen Vorteilen zählt die erweiterte Funktion des AUTO-Befehls zur automatischen Erzeugung der Zeilennummern bei der Programmierung. Während Besitzer eines CPC 464 bislang mit AUTO nur komplette Neueingaben von Basic-Zeilen vornehmen konnten, haben Benutzer des Basic 1.1 auch die Wahl, Veränderungen innerhalb bereits bestehender Zeilen durchzuführen. AUTO Plus schlichtet nun diesen Bruderzwist, indem es dem ältesten CPC-Familienmitglied Anschluß an die Moderne bietet. Es erzeugt den RSX-Befehl

`|AUTO,zeilennummer,schrittweite`

Übergeben Sie nur einen oder keinen der beiden Parameter, nimmt der Computer als Standardwerte jeweils 10 an. Wie bei RSX-Befehlen üblich, beginnt AUTO mit dem senkrechten Balken, den Sie durch gleichzeitigen Druck der Tasten <SHIFT> und <@> erzeugen. Nach Eingabe des Listings speichern Sie AUTO Plus bitte sicherheitshalber. Nach dem Start legen Sie die Adresse fest, ab der der Basic-Lader den nötigen Maschinencode ablegt, denn dieser ist im Speicher frei verschiebbar und arbeitet so mit praktisch jeder anderen Erweiterung problemlos zusammen. AUTO Plus eignet sich auch vorzüglich dazu, mit »Explora« eingegebene Listings nachträglich auf Tippfehler zu untersuchen.

(Stefan Aust/ja)

| Steckbrief | |
|---|---|
| Programm: | AUTO Plus |
| Computer: | CPC 464 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

```
10 '************************************     [D224]
20 '* AUTO+  (C) by Stefan M. Aust   *       [D5D2]
30 '*         Version 1 vom 08.3.86  *       [DFD8]
40 '************************************     [A22A]
50 DATA 01,&0009,21,&0013,C3,D1,BC,&000E
   ,C3,&0017,41,55                          [83BC]
60 DATA 54,CF,00,00,00,00,00,FE,03,D0,21
   ,0A,00,11,0A,00                          [90DE]
70 DATA FE,00,28,14,FE,01,28,0A,DD,5E,00
   ,DD,56,01,DD,23                          [D6F2]
80 DATA DD,23,DD,6E,00,DD,66,01,CD,00,B9
   ,ED,53,1F,AC,CD                          [61A0]
90 DATA D6,C0,CD,&0050,30,06,3A,1C,AC,B7
   ,20,F5,C3,64,C0                          [43D0]
100 DATA CD,D3,C0,ED,5B,1D,AC,CD,A3,E7,3
    8,16,2A,1D,AC,CD                        [ED4A]
110 DATA 79,EE,CD,11,C1,D0,7E,B7,28,06,C
    D,D2,E6,CD,7A,C1                        [1DF6]
120 DATA 37,C9,CD,63,E1,CD,43,CA,D0,CD,B
    C,E6,D2,C2,C0,C4                        [B11E]
130 DATA 7A,C1,ED,5B,1D,AC,C3,20,C1         [0CFE]
140 DATA *ENDE*                             [79BE]
150 PEN 1:PAPER 0                           [3ED8]
160 INPUT"Ladeadresse:_",Ladr               [1260]
170 MEMORY Ladr-1                           [4F24]
180 adr=Ladr                                [4660]
190 READ wert$                              [AEB8]
200 IF wert$="*ENDE*"THEN 300               [D234]
210 IF ASC(wert$)<48 THEN 240               [543A]
220 POKE adr,VAL("&"+wert$)                 [E82A]
230 GOTO 280                                [D350]
240 wort=VAL(wert$)+Ladr                    [450E]
250 POKE adr,wort AND 255                   [68C8]
260 adr=adr+1                               [E87E]
270 POKE adr,INT(wort/256)AND 255           [109C]
280 adr=adr+1                               [0E82]
290 GOTO 190                                [565C]
300 PRINT:PRINT"Laenge des Programms";ad
    r-Ladr;"Bytes."                         [C642]
310 CALL Ladr:END                           [9208]
```

**Listing. Mehr Eingabekomfort mit »AUTO Plus«**

# Perspektiven mit Tiefen

**Kein Computerfreak kann sich der Faszination dreidimensionaler Computergrafik entziehen. Die Grundlagen solcher 3D-Grafikprogramme finden Sie hier.**

Spätestens seit die ARD ihre computeranimierte »1« über bundesdeutsche Bildschirme flimmern läßt und in mehreren Fernsehsendungen Spitzenprodukte amerikanischer Computergrafik Begeisterung weckten, wurde vielen klar, was Computer in diesem Spezialbereich Grandioses leisten. Da fliegen Bücher wie Tauben durch die Luft oder man erlebt eine Reise durch die Ölkanäle eines Motors. All das ist so realistisch dargestellt, daß man seinen Augen kaum traut.

Selbstverständlich steckt hinter jedem solcher Filme ein Supercomputer, der trotz seiner enormen Geschwindigkeit für jedes einzelne Bild bis zu 20 Minuten Rechenzeit benötigt. Für den CPC-Besitzer bleiben solche Darstellungen natürlich reines Wunschdenken. Aber Spiele wie »Starion« oder »Elite« beweisen, daß sich dreidimensionale Grafik – zumindest in sogenannten »Drahtmodellen« – durchaus realisieren läßt. Mit einem professionellen Hilfsprogramm wie beispielsweise dem »CPC-Vektor« (Testbericht in Happy-Computer, Ausgabe 11/86) kann man sogar von Basic aus in diese Welt einsteigen. »Aber wie funktioniert das alles eigentlich?« ist die Frage vieler Computerbesitzer, die der Sache auf den Grund gehen wollen.

Der Grundgedanke all solcher Grafiken ist immer wieder die Mathematik. Mit Hilfe ihrer Formeln ist eine imaginäre Welt in Form von 3D-Koordinaten im Computer gespeichert. Und diese gilt es, auf dem Bildschirm sichtbar zu machen. Jeden Punkt einzeln zu berechnen ist natürlich unmöglich und so beschränkt man sich auf die Eckpunkte der dargestellten Flächen. Diese Punkte stellen das gewünsche Bild als »Drahtmodell« problemlos dar. Professionelle Programmierer geben sich damit zwar nicht zufrieden, aber Routinen zum Erkennen verdeckter Linien und zum Füllen von Flächen sind kompliziert und sehr langsam. Die Arbeitsgeschwindigkeit des Programms würde gegen Null absinken. Konzentrieren wir uns also auf die Berechnung und Darstellung von Vektorgrafiken.

## 1, 2, 3 Grafik

Jeder dreidimensionalen »Welt« liegt ein Koordinatensystem mit drei Achsen zugrunde. Bild 1 zeigt das am meisten gebrauchte kartesische Koordinatensystem. Bei ihm stehen drei Achsen, die mit x, y und z bezeichnet werden, senkrecht aufeinander. Der Schnittpunkt der Achsen heißt »Ursprung«. Seine Koordinaten sind somit immer (0,0,0). Jeder Eckpunkt einer Grafik kann in bezug auf diesen Ursprung mit Koordinaten eindeutig festgelegt werden. So beschreiben die Koordinaten (2,-1,-3) den Punkt P aus Bild 2. Das bedeutet nichts anderes, als daß Sie vom Ursprung 2 Einheiten in x-Richtung, dann 1 in -y-Richtung und zum Schluß -3 Einheiten in z-Richtung gehen. Die nächste Problematik ist vom Filmtechnischen her bekannt. Das Aufnahmeobjekt ist vorhanden, doch wo steht die Kamera und wohin ist sie gerichtet? Als Kamerastandpunkt wählen wir beispielsweise den Punkt K an der Position (-2,1,3). Der Blick der Kamera ist genau auf Punkt P ausgerichtet. In Bild 2 finden Sie sowohl Punkt K wie auch den Blickvektor $\vec{v}$ eingezeichnet. Dieser Vektor beginnt in (-2,1,3) und führt nach (2,-1,-3). Falls wir später den Blickwinkel der Kamera ändern wollen, legen wir den Endpunkt des Blickvektors einfach in einen anderen Punkt, beispielsweise nach (10,2,-5).

Doch zurück zu unserem Bild. Die von uns gewählten Koordinaten erweisen sich aber als sehr ungünstig für unsere Zwecke. Einfacher läßt es sich arbeiten, wenn wir den Standpunkt der Kamera und den Ursprung des Koordinatensystems in einen Punkt legen. Das ist sehr einfach: Wir verschieben die ganze

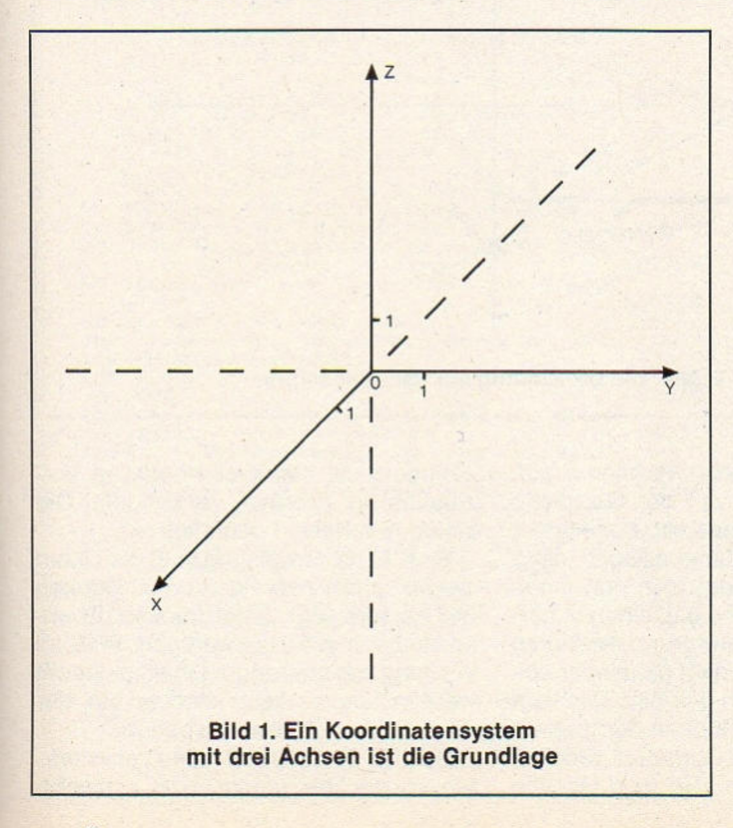


Bild 1. Ein Koordinatensystem mit drei Achsen ist die Grundlage

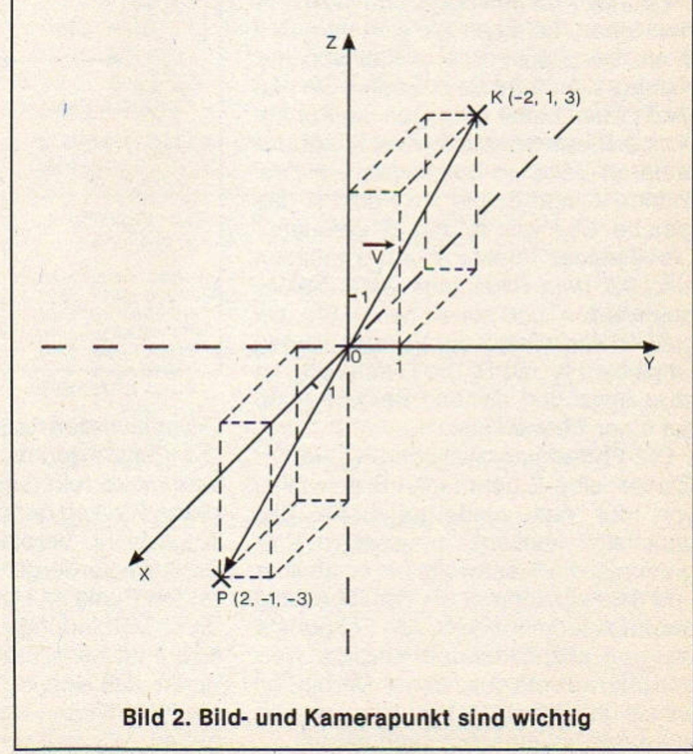


Bild 2. Bild- und Kamerapunkt sind wichtig

»Welt« und damit unsere beiden Punkte um den Wert 2 in x-, um den Wert -1 in y- und um den Wert -3 in z-Richtung. Zu den alten Koordinaten zählen wir dann die Verschiebung dazu. Der Punkt K bekommt damit den Wert (0(=-2+2), 0(=1-1), 0(=3-3)) und P den Wert (4, -2,-6). Das neue System zeigt Bild 3.

Unsere Kamera ist nun immer noch direkt auf den Punkt P (im neuen System P' genannt) gerichtet. Da der Bildschirm aber nur zwei Dimensionen darstellen kann, müssen wir die Blickrichtung entlang einer Achse festlegen. Wir wählen die z-Achse. Jede andere beliebige Achse eignet sich zwar ebenso, aber da man üblicherweise in der Ebene mit x- und y-Koordinaten arbeitet, bleiben wir auch dabei. Mit zwei Drehungen, einmal um die x- und einmal um die y-Achse, richten wir die Kamera so ein, daß sie entlang der z-Achse »schaut«. Zwei Sinusfunktionen verändern die Koordinaten von P' so, daß er zu der neuen Blickrichtung paßt.

Einen Punkt kann man nun noch nicht dreidimensional sehen, da er ja keine Ausdehnung besitzt – zumindest mathematisch betrachtet. Erst die Darstellung von zwei Punkten bewirkt einen dreidimensionalen Effekt. Dieser beruht nun einfach in der vom Blickwinkel abhängigen unterschiedlichen Entfernung, sowie der Winkellage der verbindenden Strecke zu der Ursprungsebene. Diese Effekte werden allein durch die beiden Sinusfunktionen auf die Ebene zurückgeführt. Wie das im einzelnen funktioniert, wurde bereits einmal im 2. Schneider-Sonderheft von Happy-Computer (Sonderheft 1/86) besprochen.

Um das Problem noch einmal zu verdeutlichen, schauen Sie sich bitte Bild 4 an. Der Punkt A ist weiter von der Kamera K entfernt als B. Stellen Sie nun eine Glasscheibe zwischen die Punkte A und B auf der einen und K auf der anderen Seite, so bekommen sie zwei Punkte A' und B', die am Punkt K das gleiche Bild wie A und B erzeugen. Diese beiden Punkte A' und B' müssen also auf dem Bildschirm wiedergegeben werden, und schon haben Sie das gleiche Bild wie bei der Vorlage mit den Originalen A und B. Die Darstellung ist aber einfacher, da jetzt beide Punkte auf einer Ebene liegen.

Die Entfernung zwischen K und der Glasscheibe E nennt man Brennweite, von der das wiedergegebene Bild ursächlich abhängt. In unserem Programm ist die Brennweite frei einstellbar und deshalb können Sie damit beliebig herumexperimentieren. Alle Eckpunkte unseres abzubildenden Körpers werden jetzt korrekt gezeichnet. Verbinden wir sie untereinander mit Linien, so entsteht das gewünschte Drahtmodell.

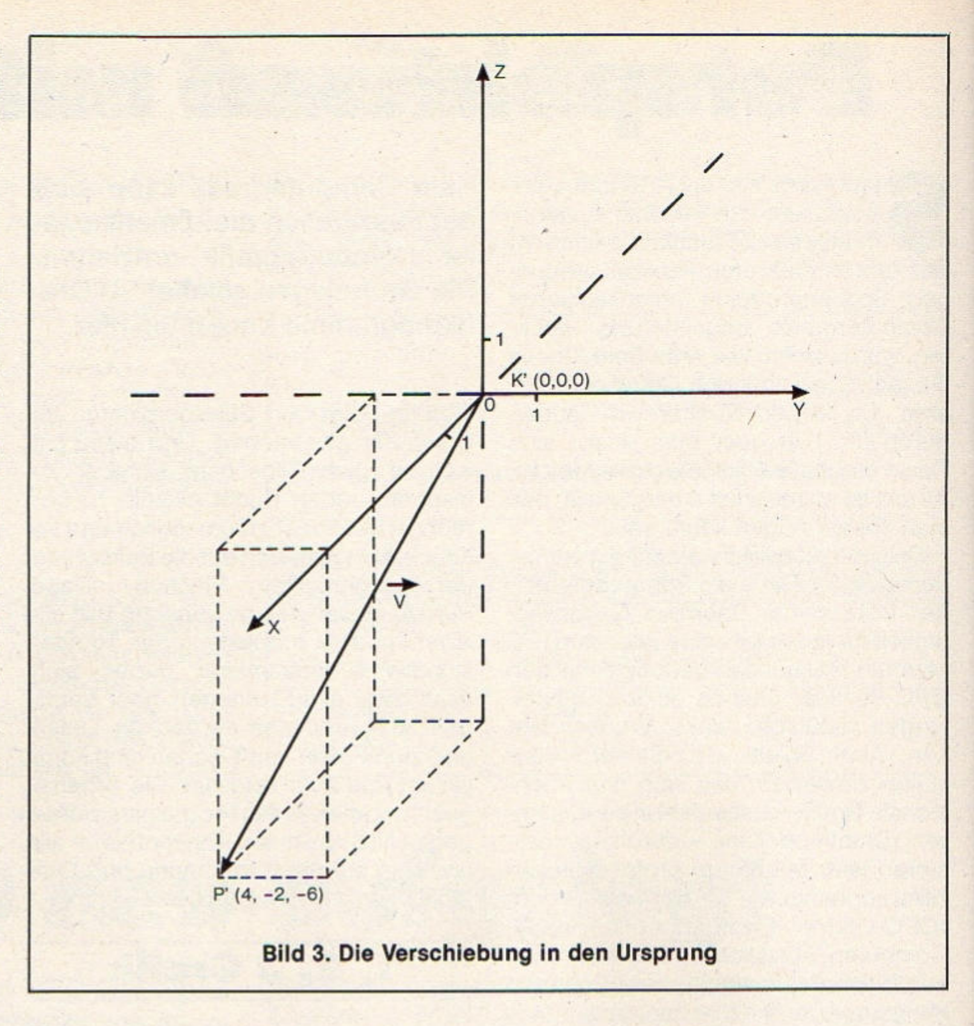



**Bild 3. Die Verschiebung in den Ursprung**

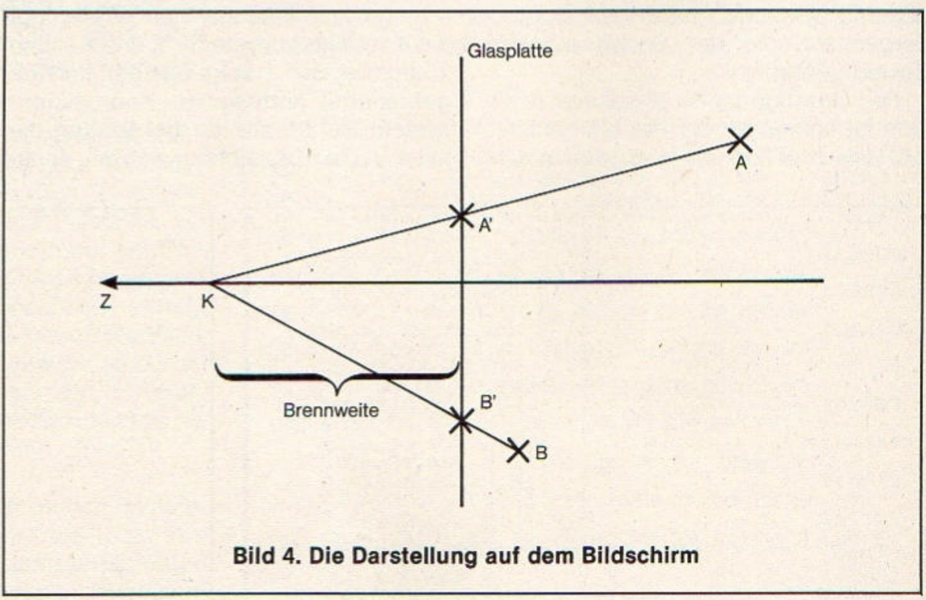



**Bild 4. Die Darstellung auf dem Bildschirm**

Dabei treten jedoch Probleme auf. Die Punkte rechts von der Glasplatte werden korrekt dargestellt. Diejenigen, die im Rücken der Kamera liegen, allerdings nicht. Verbindet man nun einen Punkt im Vordergrund und einen unkorrekten Punkt im Hintergrund, so laufen diese Verbindungslinien scheinbar völlig unmotiviert durch das Bild. Das liegt daran, daß sich im Rücken der Kamera die Koordinaten ins Gegenteil verkehrt haben. Wir müssen also eine Routine einfügen, die eventuelle negative Vorzeichen in positive verwandelt. Die Linien erscheinen dann korrekt.

Eine letzte Anweisung muß die Linien die zwischen zwei Punkten im Rücken der Kamera liegt, unterdrücken. Ebenso müssen die rückwärtigen Teile im Vordergrund startender Linien gelöscht werden. Denn diese sind ja auf der Glasplatte auch nicht zu sehen.

Listing 1 zeichnet eine stilisierte Maske, die über einem Gitter schwebt.





Das Gitter ersetzt den Boden. Die Daten für die Maske und das Gitter sind in den Zeilen 70, 71 und 72 abgelegt. Zeile 20 initialisiert das Programm und legt den Ursprung in die Mitte des Bildschirms. Zeile 40 legt die Zahl der einzulesenden Punkte und die der Linien fest. In Zeile 50 werden die Original- und die Bildschirmpunkte in Felder eingelesen, bevor im Anschluß die Masken- und Gitterwerte gelesen werden. Nach der Definition liegt nun sowohl die Maske wie auch das Gitter im Ursprung. Das ist natürlich unerwünscht und so verschieben wir das Gitter einfach linear um 50 Punkte nach unten. Dies ist eine sehr einfache Methode, ein Bild in der Normalebene einzugeben und erst später an seinen tatsächlichen Platz zu verschieben. Es ist nämlich bedeutend einfacher, ein Bild um den Ursprung herum zu zeichnen, als es direkt zu berechnen.

Auch unser Gitter soll noch manipuliert werden. Da es zu klein ist, multiplizieren wir es beim Einlesen mit dem Faktor 5. In Zeile 90 wird die Brennweite und der Beobachtungspunkt festgelegt, der im Moment der Ursprung ist. Dort haben wir ja unsere Maske definiert. Den Kamerapunkt enthält Zeile 110.

Ab Zeile 150 beginnen unsere oben erwähnten Umformungen – ab Zeile 170 die Verschiebung und ab 200 die Drehung. In Zeile 270 wird um die x- und in Zeile 280 um die y-Achse gedreht. Zeile 315 testet auf unzulässige Punkte (im Rücken der Kamera), und Zeile 320 bringt das Ergebnis auf den Bildschirm. In Basic ist das Programm zwar langsam, aber das Ergebnis kann sich sehen lassen.

## Aktion in der zweiten Bank

Doch unser Programm kann noch mehr. Ändern Sie die in Listing 2 angegebenen Zeilen, so fährt die Kamera von unten rechts nach oben links, den Blick immer auf die Maske gerichtet. Für den CPC 464 ändern Sie dazu in den Zeilen 140, 1090 und 1100 die Adresse B7C6 in B1CB hex. Damit Sie während der Berechnung nicht vor einem leeren Bildschirm sitzen, arbeiten wir mit zwei Speicherbereichen zur Bildausgabe. Der zweite Bereich steht ab der Adresse 4000 hex.

Listing 3 dreht die Maske während der Kamerafahrt um die y-Achse. Falls einmal die Körperachse eines Objekts nicht mit der Drehachse übereinstimmt, dann verschieben Sie nur das Objekt so, bis beide Achsen zur Deckung kommen. Dort drehen Sie es, bevor Sie es wieder an seinen Platz zurückschieben.

Was jetzt vor Ihnen abläuft, ist ein (langsamer) 3D-Film. Viele sind durch all die mathematischen Erklärungen sicher noch etwas verwirrt, aber eins hilft immer: Listing abtippen und ausprobieren. (Oliver Hansen/ja/hg)

| Steckbrief | |
|---|---|
| Programm: | 3D-Grafik |
| Computer | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Diskette/Kassette |

```
10 '3d-Darstellung                          [63E2]
20 ON ERROR GOTO 5000:MODE 2:PAPER 1:PEN
   0:CLS:DEG:ORIGIN 320,200                 [3A86]
30 'Daten der Figur einlesen                [DD02]
40 n%=48:lines%=51:  'Anzahl der Punkte,
   Anzahl der Linien                        [F6A4]
50 DIM x(n%),y(n%),z(n%),x2(n%),y2(n%),z
   2(n%)                                    [7F3E]
60 RESTORE 70:FOR i%=1 TO 16:READ x(i%),
   y(i%),z(i%):x(i%)=x(i%)-37.5:z(i%)=z(
   i%)-8.3:NEXT i%:' Maske einlesen und
   zentrieren                               [D8EA]
65 FOR i%=17 TO n%:READ x(i%),y(i%),z(i%
   ):x(i%)=x(i%)*5:z(i%)=z(i%)*5:y(i%)=y
   (i%)-50:NEXT i%:' Gitter einlesen, ve
   rgroessern und verschieben               [13CC]
70 DATA 37.5,100,0,0,87.5,0,75,87.5,0,16
   .6,83.3,13,58.3,83.3,13,37.5,74,10,7.
   5,58.3,8,67.5,58.3,8,37.5,37.5,16.6,1
   8.7,31.2,10,56.3,31.2,10,37.5,15,13,1
   5.6,9.4,6.2,59.4,9.4,6.2,9.4,0,0,65.6
   ,0,0:' Punkte Maske                      [B034]
71 DATA -40,0,-40,-30,0,-40,-20,0,-40,-1
   0,0,-40,0,0,-40,10,0,-40,20,0,-40,30,
   0,-40,40,0,-40,-40,0,40,-30,0,40,-20,
   0,40,-10,0,40,0,0,40,10,0,40,20,0,40,
   30,0,40,40,0,40,-40,0,-30,-40,0,-20,-
   40,0,-10,-40,0,0,-40,0,10,-40,0,20,-4
   0,0,30,40,0,-30,40,0,-20                 [BC00]
72 DATA 40,0,-10,40,0,0,40,0,10,40,0,20,
   40,0,30: 'Punkte Gitter                  [77F4]
80 'Festlegung von Brennweite und Blickv
   ektor                                    [13E0]
90 bw=400:vx=0:vy=0:vz=0                    [F3EA]
100 'Festlegung der Kameraparameter         [7EF2]
110 ox=0:oy=0:oz=250                        [7B18]
130 GOSUB 150                               [8ED4]
140 END                                     [9D18]
150 '********** Transformation der Koord
    inaten ****************                 [4E0E]
160 ' 1. Verschiebung, so dass ox,oy,oz
    den Ursprung bildet                     [A9F8]
170 vx2=vx-ox:vy2=vy-oy:vz2=vz-oz           [AEA4]
180 FOR i%=1 TO n%:x2(i%)=x(i%)-ox:y2(i%
    )=y(i%)-oy:z2(i%)=z(i%)-oz:NEXT i%      [3F46]
190 ' 2. Drehung, so dass die Blickricht
    ung genau auf der Z-Achse liegt         [E952]
200 betrag=SQR(vz2^2+vy2^2+vx2^2)           [2A28]
210 sn1=vy2/betrag:cs1=-vz2/betrag          [0470]
220 IF vy2=0 THEN sn1=0:cs1=1               [4E1C]
230 t=vz2*cs1-vy2*sn1:vy2=vz2*sn1+vy2*cs
    1:vz2=t                                 [2C86]
240 betrag=SQR(vx2^2+vz2^2)                 [B178]
250 sn2=vx2/betrag:cs2=vz2/betrag           [1020]
260 FOR i%=1 TO n%                          [4762]
270 t=z2(i%)*cs1-y2(i%)*sn1:y2(i%)=z2(i%
    )*sn1+y2(i%)*cs1:z2(i%)=t:' Drehung
    um die x-Achse                          [6710]
280 t=x2(i%)*cs2-z2(i%)*sn2:z2(i%)=x2(i%
    )*sn2+z2(i%)*cs2:x2(i%)=t:' Drehung
    um die y-Achse                          [B516]
285 IF z2(i%)<0 THEN y2(i%)=-y2(i%):x2(i
    %)=-x2(i%):' Punkt hinter Kamera        [C06C]
290 NEXT i%                                 [CC50]
295 IF flag=1 THEN flag=0:GOTO 1090 ELSE
     flag=1:GOTO 1100                       [5620]
300 CLS:RESTORE 1000:FOR i%=1 TO lines%     [5768]
310 READ a%,b%:' Anfangs- und Endpunkt      [62CA]
315 IF z2(a%)<0 AND z2(b%)<0 THEN 330       [4CBE]
320 MOVE (bw*x2(a%))/z2(a%),(bw*y2(a%))/
    z2(a%):DRAW (bw*x2(b%))/z2(b%),(bw*y
    2(b%))/z2(b%),0                         [F6D4]
330 NEXT i%:RETURN                          [EF7A]
1000 DATA 1,2,2,15,15,16,16,3,3,1,1,4,4,
     6,6,5,5,1,2,4,3,5,6,7,7,4,6,8,8,5,6
     ,9,9,10,10,6,6,11,11,9,10,11,7,10,8
     ,11,10,13,13,12,12,14,14,11,11,12,1
     2,10,15,13,13,14,14,16,1,6             [41C0]
1001 DATA 17,25,25,34,34,26,26,17,18,27,
     19,28,20,29,21,30,22,31,23,32,24,33
     ,35,42,36,43,37,44,38,45,39,46,40,4
     7,41,48                                [B85A]
5000 RESUME NEXT                            [632A]
```

**Listing 1. Die dreidimensionale »Maske« entsteht über dem Gitter**

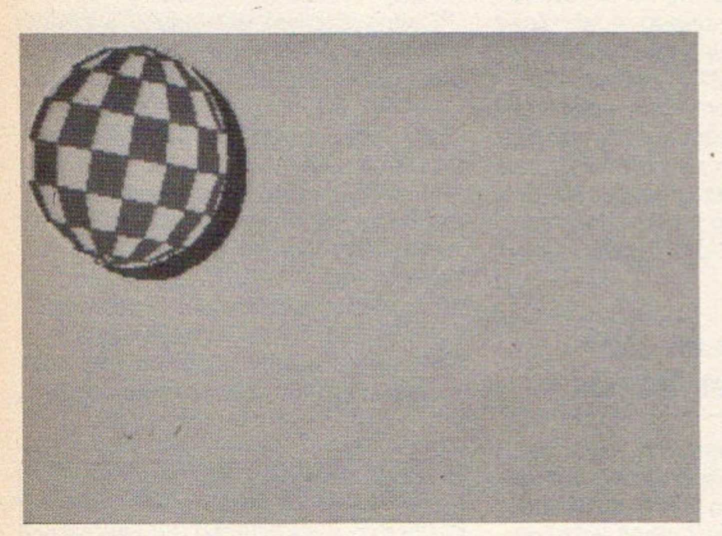
```
20 MEMORY &4000:ON ERROR GOTO 5000:MODE
   2:PAPER 1:PEN 0:CLS:DEG:ORIGIN 320,20
   0                                        [3AC0]
110 ox=-210:oy=100:oz=250                   [13FA]
120 FOR oy=-400 TO 400 STEP 20              [0AEA]
125 ox=ox+10                                [029E]
130 GOSUB 150                               [8ED4]
140 NEXT:OUT &BD00,48:POKE &B7C6,192:MOD
    E 2:END                                 [7D2A]
295 IF flag=1 THEN flag=0:GOTO 1090 ELSE
     flag=1:GOTO 1100                       [5620]
1090 OUT &BC00,12:OUT &BD00,16:POKE &B7C
     6,192:GOTO 300                         [0C38]
1100 OUT &BC00,12:OUT &BD00,48:POKE &B7C
     6,64:GOTO 300                          [F2CE]
```

**Listing 2. Eine »Kamerafahrt« durch den Raum**

```
185 GOSUB 1040                              [EB46]
1040 ' Drehung um die Y-Symmetrieachse      [616C]
1045 cs3=COS(15):sn3=SIN(15):FOR i%=1 TO
      16                                    [86FE]
1060 t=x(i%)*cs3-z(i%)*sn3:z(i%)=x(i%)*s
     n3+z(i%)*cs3:x(i%)=t                   [F388]
1080 NEXT i%:RETURN                         [76E0]
```

**Listing 3. Die drehende Maske ist 3D-perfekt**

# Der Amiga-Ball springt



**Nach vielen Umsetzungen dieser beliebten Grafik-Demonstration für diverse Computer haben nun auch Sie als CPC-Besitzer Grund zur Freude.**

Wer die bekannte Grafikdemo des Commodore-Amiga je gesehen hat, den verblüfft wahrscheinlich der karierte Hüpfball auf seinem Schneider CPC. Die hohe Geschwindigkeit dieser Grafikanimation läßt sich auf den Schneider-Computern natürlich nur durch Tricks realisieren. So baut Listing 2 nacheinander acht Einzelbilder mit Bewegungsphasen des sich drehenden Balls auf. Da dieser Vorgang sehr zeitraubend ist (zirka 15 Minuten), speichert die Routine die Teilbilder unter dem Namen »AMIGA.PIC« gemeinsam in einer Datei.

Für den Lauf des Bildgenerators ist eine Maschinencode-Routine notwendig. Sie erzeugt der Basic-Lader in Listing 1 und speichert sie unter dem Namen »AMIGA.BIN« als Binärdatei. Derselbe Maschinencode enthält auch die Routine zur Animation des Balls. Listing 3 dient ausschließlich dem späteren Laden der beiden benötigten Binärdateien und startet dann die Endlosschleife, innerhalb der der Ball sich »bewegt«. Der Abbruch ist nur durch einen Reset möglich. Kassetten-Benutzer speichern für das lauffähige Endprodukt hintereinander zuerst diese Laderoutine, danach den Maschinencode »AMIGA.BIN« und als letztes die Bilder »AMIGA.PIC«, um das automatische Nachladen sicherzustellen.

(Ralf Brinkmann/ja)

```
100 '************************************    [A480]
101 '* AMIGA.DAT - DATA-Lader von 'CPC' *    [C74C]
102 '************************************    [1C84]
103 '                                        [DEB6]
104 DATA 9C40,21,00,C0,22,58,98,21,98,0ADA  [313A]
105 DATA 9C48,3A,22,5A,98,3E,01,32,62,16B2  [3248]
106 DATA 9C50,98,32,60,98,32,64,98,CD,44FD  [C288]
107 DATA 9C58,68,9C,CD,E1,9C,CD,27,9D,03B7  [0D26]
108 DATA 9C60,3A,F4,B4,FE,80,C8,18,EF,3E9F  [0F3C]
109 DATA 9C68,3A,66,98,FE,01,CA,8F,9C,1AC2  [DA18]
110 DATA 9C70,3A,62,98,FE,31,C2,7E,9C,1B80  [BABE]
111 DATA 9C78,3E,01,32,66,98,C9,3C,32,18CE  [488E]
112 DATA 9C80,62,98,2A,58,98,CD,20,BC,10C8  [6DA4]
113 DATA 9C88,22,58,98,CD,AE,9C,C9,3A,1E78  [680E]
114 DATA 9C90,62,98,FE,01,C2,9D,9C,3E,0DB2  [87E8]
115 DATA 9C98,00,32,66,98,C9,3D,32,62,0F7A  [3774]
116 DATA 9CA0,98,2A,58,98,CD,23,BC,22,43BE  [44E0]
117 DATA 9CA8,58,98,CD,C6,9C,C9,2A,5A,180A  [751A]
118 DATA 9CB0,98,3E,8B,BC,DA,BF,9C,01,5C35  [5140]
119 DATA 9CB8,B8,0B,09,22,5A,98,C9,21,5CC3  [6DC8]
120 DATA 9CC0,98,3A,22,5A,98,C9,2A,5A,448A  [FBBC]
121 DATA 9CC8,98,3E,3B,BC,D2,DA,9C,01,4BE1  [EB30]
122 DATA 9CD0,B8,0B,EE,00,ED,42,22,5A,457E  [FBDE]
123 DATA 9CD8,98,C9,21,A0,8C,22,5A,98,74A4  [47B6]
124 DATA 9CE0,C9,3A,64,98,FE,01,CA,08,6968  [53DC]
125 DATA 9CE8,9D,3A,60,98,FE,01,C2,F7,4307  [1DD6]
126 DATA 9CF0,9C,3E,01,32,64,98,C9,2A,4278  [5494]
127 DATA 9CF8,58,98,CD,29,BC,22,58,98,1470  [0CBA]
128 DATA 9D00,3A,60,98,3D,32,60,98,C9,1439  [A352]
129 DATA 9D08,3A,60,98,FE,63,C2,16,9D,1941  [349C]
130 DATA 9D10,3E,00,32,64,98,C9,2A,58,18E8  [6C64]
131 DATA 9D18,98,CD,26,BC,22,58,98,3A,713A  [47B0]
132 DATA 9D20,60,98,3C,32,60,98,C9,2A,1278  [2D4A]
133 DATA 9D28,58,98,22,5C,98,2A,5A,98,0FC4  [0FAC]
134 DATA 9D30,22,5E,98,06,64,C5,ED,5B,1455  [2A88]
135 DATA 9D38,5C,98,2A,5E,98,01,1E,00,0C58  [8F8A]
136 DATA 9D40,ED,B0,2A,5C,98,CD,26,BC,5D04  [C212]
137 DATA 9D48,22,5C,98,2A,5E,98,01,1E,172C  [3392]
138 DATA 9D50,00,09,22,5E,98,C1,10,DD,04D9  [B374]
139 DATA 9D58,C9,00,21,00,C0,22,5A,98,6604  [4634]
140 DATA 9D60,DD,6E,00,DD,66,01,22,58,7BF8  [81BC]
141 DATA 9D68,98,CD,6D,9D,C9,2A,58,98,7DF8  [8B38]
142 DATA 9D70,22,5C,98,2A,5A,98,22,5E,170A  [407E]
143 DATA 9D78,98,06,64,C5,ED,5B,5C,98,4B74  [94DA]
144 DATA 9D80,2A,5E,98,01,1E,00,ED,B0,100A  [308C]
145 DATA 9D88,2A,5C,98,01,1E,00,09,22,11D0  [6446]
146 DATA 9D90,5C,98,2A,5E,98,CD,26,BC,0FA4  [2E1A]
147 DATA 9D98,22,5E,98,C1,10,DD,C9,00,1BF6  [F1CE]
148 DATA *ENDE*                              [89CE]
149 adr=&9C40:zeile=104                       [46BA]
150 MEMORY adr-1                             [D588]
151 READ d$                                  [34F6]
152 IF d$="*ENDE*"THEN 165                   [CF96]
153 pr=0                                     [4E10]
154 FOR i=1 TO 8                             [3366]
155 READ a$:a=VAL("&"+a$)                    [2844]
156 POKE adr,a:adr=adr+1                     [B720]
157 pr=pr*2:IF pr>65535 THEN pr=pr-65535     [06A0]
158 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553
    6                                        [CFB8]
159 NEXT i                                   [430E]
160 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN
     pr2=pr2+65536                           [6888]
161 IF pr<>pr2 THEN 164                      [76C2]
162 zeile=zeile+1                            [CA08]
163 GOTO 151                                 [FA54]
164 PRINT"Pruefsummenfehler in Zeile";zeile
    :STOP                                    [73FA]
165 SAVE"AMIGA.BIN",b,&9C40,&160,&9C40:END   [0B02]
```

**Listing 1. Im Basic-Lader ist der komplette Maschinencode zur Bewegung des Amiga-Balls enthalten**

```
10 ' //////////////////////         [A5A4]
20 ' ///                ///         [BBC6]
30 ' ///   Amiga-Ball   ///         [8256]
40 ' /// (c)    1986    ///         [CB22]
50 ' /// Ralf Brinkmann ///         [BDD6]
60 ' ///                ///         [2BCE]
70 ' //////////////////////         [F5B0]
80 MEMORY &3A97:LOAD"amiga.bin":xorg=10:
   yorg=220                         [AEE6]
90 INK 0,13:INK 1,26:INK 2,0:INK 3,6:BOR
   DER 14                           [5FC8]
100 MODE 1:z=0:adr=15000            [7FD8]
110 ORIGIN xorg,yorg                [268E]
120 GOTO 420                        [BA44]
130 h=COS (w)*SIN (k*p)             [7160]
140 y=SIN (w)                       [887A]
150 x=h*cw-y*sw                     [D022]
160 y=h*sw+y*cw                     [1122]
170 x=n+xd+m+m*x-XPOS                        [9D56]
180 y=  yd-m+m*y-YPOS                        [88B2]
190 RETURN                                   [9434]
200 FOR n=0 TO 7                             [2E5C]
210 v=n/4                                    [E46E]
220 GOSUB 290                                [43DE]
230 FOR k=-3.6 TO 3.6 STEP 0.05              [13C2]
240 FOR f=-4 TO 3                            [0AAE]
250 u=INT (K+v)+f                            [F184]
260 IF u=2*INT (u/2) THEN LET w=f*p:PLOT
    R 0,0:GOSUB 130:PLOTR x,y,3:w=w+p:GO
    SUB 130:DRAWR x,y:DRAWR -x,-y,3          [D460]
270 NEXT f:NEXT k                            [3204]
280 NEXT n                                   [450E]
290 xd=20:yd=156                             [674C]
300 GOSUB 340:PRINT CHR$(22);CHR$(1);        [BF00]
310 xd=2:yd=166                              [50E0]
320 GOSUB 340:PRINT CHR$(22);CHR$(0);        [9B02]
```

```
330 RETURN                                        [A62C]
340 z=z+1                                         [DF88]
350 IF z MOD 2=0 THEN PLOT 500,500,1 ELS
    E PLOT 500,500,2                              [753A]
360 IF z MOD 2=1 AND z>2 THEN GOSUB 440           [2E8C]
370 FOR r=-m TO m STEP 2                          [E310]
380 kr=SQR (m*m-r*r)                              [4DF6]
390 PLOT xd+m+n+r,yd-m+kr                         [65E0]
400 DRAWR 0,-2*kr:NEXT r                          [D742]
410 RETURN                                        [AB2A]
420 m=90:p=-PI/20:sw=SIN (p):cw=COS (p):
    p=PI/8                                        [7E62]
430 GOTO 200                                      [4144]
440 ORIGIN 0,201:DRAW 600,0,0                     [72CA]
450 CALL 40282,adr:adr=adr+3000:ORIGIN x
    org,yorg:xorg=xorg-1                          [719E]
460 PLOT 0,0,2:CLS                                [833E]
470 IF adr>=39000 THEN 490                        [9A48]
480 RETURN                                        [1D38]
490 SAVE"amiga.pic",b,15000,24000                 [5FD6]
500 CALL 40000:END                                [09EC]
```

**Listing 2. Nach etwa 15 Minuten speichert der Generator die acht Teilbilder**

| Steckbrief | |
|---|---|
| Programm: | Amiga-Ball |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora/CPC |
| Datenträger: | Kassette/Diskette |

```
10 MEMORY &3A97                                   [CB08]
20 LOAD"AMIGA.PIC":LOAD"AMIGA.BIN"                [7DA6]
30 INK 0,13:INK 1,26:INK 2,0:INK 3,6:BOR
   DER 14                                         [39BC]
40 MODE 1:CALL &9C40                              [96EC]
```

**Listing 3. Wenn Sie diese Zeilen starten, läuft der Rest der Grafikdemo vollautomatisch**

# Software-Glück

**»Soft-Chef« gehört zu den Strategie- und Simulationsspielen. Mit Geschick und einer Portion Glück führen Sie Ihre Software-Firma zum Erfolg.**

Als Manager einer Software-Firma müssen Sie ein Jahr lang Ihre Programme erfolgreich verkaufen, um damit möglichst viel Geld zu verdienen und eventuell sogar den Software-Cup zu erringen – eine hohe Auszeichnung der Software-Branche. In diesem Bestreben konkurrieren mit Ihnen neun andere, vom Computer gesteuerte Unternehmen. Jede der zehn Firmen besitzt die Rechte an jeweils zehn verschiedenen Programmen, die Sie möglichst erfolgreich vermarkten sollen. Woche für Woche ermittelt der Computer die Verkaufszahlen aller Programme und errechnet daraus die »Top 18« – eine Rangliste der 18 meistverkauften Produkte. Nur Firmen, deren Programme dort vertreten sind, ernten Prämien und Punkte für den Software-Cup; die Höhe richtet sich nach der jeweiligen Plazierung.

Zusätzlichen Gewinn für plazierte Programme verbuchen Sie, falls der Verkaufspreis über der Gewinngrenze liegt. Das birgt aber auch Risiken: Je höher Sie den Verkaufspreis festsetzen, desto weniger Exemplare verkaufen sich. Billige Programme sind nun einmal leichter zu verkaufen. Der Computer limitiert allerdings Höchst- und Niedrigstpreise, um einem ruinösen Wettbewerb und Wucher vorzubeugen.

Je Programm steht Ihnen wöchentlich ein Werbeetat von 10 000 Mark zur Verfügung. Natürlich reagiert die Konkurrenz auf Ihre Preis- und Werbepolitik mit entsprechenden Maßnahmen. Ihre Verkaufszahlen richten sich also nach Verkaufspreis und Werbeaufwand, aber auch nach Qualität und Index der Produkte. Die Qualität der Programme legt der Computer zu Beginn des Spiels fest. Der Index berücksichtigt die Aktualität und steht deshalb bei Neuerscheinungen immer auf 1. Jede Woche sinkt er um 0,01 Einheiten, so daß der Verkauf im Laufe der Zeit zurückgeht. Um zu verhindern, daß Ihr Gesamtumsatz stagniert oder sich gar rückläufig entwickelt, geben Sie öfter neue Programme in Auftrag. Dadurch erreichen Sie eine Qualitätssteigerung und bereichern Ihr Angebot mit stets aktuellen Produkten. Die maximal erreichbare Qualität repräsentiert der Wert 25. Die Höhe der Entwicklungskosten richtet sich nach der Qualitäts- und Indexdifferenz zum alten Programm und nach Länge der Entwicklungszeit, die zwischen einer und zehn Wochen liegt (je schneller, desto teurer).

SOFT - CHEF 2.9 / (C) 1986 by MARTIN STAHL / RUESTA

Woche : 2 — Die Hitliste der 18 bestverkauften Programme dieser Woche :

| | Firma : | Titel : | Anzahl : | Punkte : | |
|---|---|---|---|---|---|
| 1. | TomSoft | Pitstart II | 3303 | 30 | <<<<< |
| 2. | TomSoft | Facing destr. Set | 3292 | 25 | <<<<< |
| 3. | Sublogik | Sommer Games II | 3272 | 20 | |
| 4. | TomSoft | Imbossible Mission | 3265 | 15 | <<<<< |
| 5. | Antirock | Spass Taxi | 3212 | 14 | |
| 6. | Essix | Sommer Games | 3148 | 13 | |
| 7. | Essix | Winter Shames | 3073 | 12 | |
| 8. | TomSoft | Sublogik Light II | 3042 | 11 | <<<<< |
| 9. | Bruderbund | Master of the Champs | 2994 | 10 | |
| 10. | Sydney House | In-Fort-Tennis | 2955 | 9 | |
| 11. | Elise | Pfeifenlinie | 2914 | 8 | |
| 12. | Bruderbund | Three-on-Three | 2867 | 7 | |
| 13. | TomSoft | Arschon | 2834 | 6 | <<<<< |
| 14. | Elise | Matsch Point | 2809 | 5 | |
| 15. | Elise | Kikstop | 2791 | 4 | |
| 16. | Sydney House | Bella 7 | 2772 | 3 | |
| 17. | TomSoft | Foodball Manager | 2772 | 2 | <<<<< |
| 18. | Essix | Smokie | 2697 | 1 | |

< SPACE druecken >

**In der Hitliste sind die Produkte des Spielers rechts markiert**

Nach dem Programmstart wählen Sie einen Schwierigkeitsgrad zwischen 0 und 7. Je höher der Schwierigkeitsgrad liegt, um so geringer ist anfangs die Qualität Ihrer zehn Programme. Darauf folgt das Hauptmenü mit folgenden Unterpunkten zur Wahl:

**Werbung**

Die Höhe des Werbeetats je Woche und Produkt ist auf 10 000 Mark begrenzt.

**Neuentwicklung**

Sie wählen eines Ihrer Programme zum Austausch. Dann bestimmen Sie gewünschte Qualität und Entwicklungszeit des neuen Produkts. Aus diesen Angaben errechnet der Computer automatisch die Entwicklungskosten. Erscheint Ihnen der Aufwand lohnend, erteilen Sie den Entwicklungsauftrag.

Wollen Sie lediglich den Index eines Programms wieder erhöhen, übernehmen Sie einfach den alten Wert für die Qualität.

| Steckbrief | |
|---|---|
| Programm: | Soft-Chef |
| Computer: | CPC 464/664/6128 |
| Checksummer: | Explora |
| Datenträger: | Kassette/Diskette |

**Preise neu festsetzen**

Die Verkaufspreise Ihrer Programme variieren von 29 bis 69 Mark. Sie erzielen jedoch nur jenseits der 29-Mark-Grenze wirklich interessante Gewinne. Hier ist besonderes kaufmännisches Fingerspitzengefühl gefordert, um auf der einen Seite die Ware nicht zu »verschenken«, auf der anderen aber nicht wegen zu hoher Preise auf den Produkten sitzenzubleiben.

**Statistik**

führt Sie in ein Untermenü, wo Sie sich einen aktuellen Überblick verschaffen: Gesamtumsätze aller Firmen, Gesamt-Stückzahlen der verkauften Programme, Kontostände sämtlicher Firmen und Chancen für den Software-Cup, sowie Verteilung der Programme auf die verschiedenen Firmen und in Entwicklung begriffene eigene Programme.

**Highscores anzeigen**

Der Computer speichert selbständig für alle acht Schwierigkeitsgrade die Höchst-Punktzahlen in den Sparten: Anzahl der verkauften Exemplare eines Programms pro Woche und Gesamtzahl verkaufter Exemplare einer Firma. Zusätzlich gibt es noch Highscores für die in der Endabrechnung nach einem Jahr erhaltenen Punkte.

**Abspeichern des Spielstands**

Da ein Spieldurchgang relativ lange dauert, ist es mitunter sinnvoll, den aktuellen Spielstand zu speichern, um erst später das Spiel fortzusetzen.

**Laden des Spielstands**

Das Gegenstück zum Speichern. Diese beiden Menüpunkte bieten sich an, wenn Sie einmal »halsbrecherische« Experimente vorhaben. Treiben Sie sich damit in den Ruin, laden Sie einfach den Spielstand einer erfolgreicheren Geschäftsperiode und fangen mit dieser günstigeren Konstellation wieder an.

**Weiter im Spielverlauf**

Hier beginnt die Prozedur, die sich 52mal innerhalb eines Spieldurchgangs wiederholt. Auf die Anzeige sämtlicher Neuentwicklungen folgt eine Übersicht Ihrer Verkäufe. Dann errechnet der Computer die Rangliste der aktuellen Top 18. Es folgt der Zwischenstand im Software-Cup und eine Abrechnung Ihrer Verkaufserfolge, bevor wieder das Hauptmenü erscheint. Nach 52 Wochen bricht Soft-Chef ab und verteilt für Ihre Leistungen Punkte. Wenn Sie 1000 Punkte und mehr erzielen, dürfen Sie zufrieden sein; erst recht, wenn Sie das im siebten Schwierigkeitsgrad geschafft haben.

Ein Hinweis zur Eingabe der Listings: Geben Sie bitte Listing 1 zuerst ein und starten es. Es erzeugt die später benötigte Highscore-Datei. Danach brauchen Sie diesen Programmteil nicht mehr. Listing 2 enthält das Hauptprogramm. Kassettenbenutzer speichern bitte die Highscore-Datei unmittelbar hinter dem Programm, weil dies gleich zu Beginn den Inhalt dieser Datei benötigt. (Martin Stahl/ja)

```
1 DIM hsf(8),hsff(8),hsw(8),hswf(8),hswt
  (8),h1$(8),h2$(8),hsp(8),h3$(8)          [9672]
2 FOR t=1 TO 8                             [F9AC]
3 h1$(t)="............"                    [82FC]
4 h2$(t)="............"                    [CE00]
5 h3$(t)="............":NEXT t             [5B1E]
10 OPENOUT "softchef.hsc"                  [DD9A]
20 FOR t=1 TO 8                            [EA0C]
30 WRITE#9,hsf(t)                          [0A38]
40 WRITE#9,hsff(t)                         [D006]
50 WRITE#9,hsw(t)                          [CD5E]
60 WRITE#9,hswf(t)                         [5C2C]
70 WRITE#9,hswt(t)                         [234A]
80 WRITE#9,h1$(t)                          [B53A]
90 WRITE#9,h2$(t)                          [C53E]
92 WRITE#9,hsp(t)                          [F65C]
94 WRITE#9,h3$(t)                          [BF48]
100 NEXT t                                 [2808]
110 CLOSEOUT                               [F840]
```

**Listing 1. Dieser Programmteil erzeugt die Highscore-Datei**

```
10 REM ********************************
   ********************************
   **                                        [43AA]
20 REM *

   *                                         [9234]
30 REM *                       S O F T -
     C H E F  2.9
   *                                         [8D66]
40 REM *

   *                                         [DA38]
50 REM *                 (C)  1986 by MA
   RTIN STAHL / Ruesta
   *                                         [D6DC]
60 REM *

   *                                         [BA3C]
70 REM ********************************
   ********************************
   **                                        [67B6]
72 OPENOUT"D":MEMORY HIMEM-1:CLOSEOUT        [23C8]
75 GOSUB 59700                               [2662]
77 RANDOMIZE TIME                            [F90C]
80 MODE 2                                    [EBFE]
85 WINDOW #2,14,66,1,1:PAPER #2,13:PEN #
   2,0                                       [23C4]
86 WINDOW #0,1,80,2,25                       [8E76]
90 DIM f$(10),prg$(100),prg(100),ha(100)
   ,f(10,10),vk(10,10),gvk(10,10),pr(10,
   10),w(10,10),wsum(10,10),p(100),p1(10
   0),p2(100),pk(10),ind(100),konto(10),
   gv(100),gv1(100),gv2(100),r1(10)          [0656]
100 REM ***  Initialisierung  ***            [3B42]
110 FOR t=2 TO 10                            [FDC0]
120 READ f$(t)                               [647C]
130 NEXT                                     [61E6]
140 FOR t=1 TO 100                           [5E24]
150 READ prg$(t),prg(t)                      [C5BC]
160 NEXT                                     [75EC]
161 FOR t=1 TO 100                           [852A]
162 ind(t)=1                                 [604E]
163 NEXT t                                   [4B1A]
170 PRINT#2,"SOFT - CHEF 2.9<2>/<2>(C) 1
   986 by MARTIN STAHL / RUESTA";            [F67A]
172 GOSUB 59500                              [2EBA]
175 CLS                                      [763E]
180 LOCATE 1,6:PRINT"Bitte geben Sie den
    Namen Ihrer Software - Firma ein :
   ";                                        [41C2]
190 INPUT f$(1)                              [B9EC]
200 f$(1)=LEFT$(f$(1),20)                    [3B6A]
210 CLS                                      [782A]
220 LOCATE 1,6:PRINT"Programme werden au
   f die verschiedenen Firmen verteilt
   !"                                        [18F6]
225 FOR t1=1 TO 10                           [0C2E]
230 FOR t=1 TO 10                            [F7C4]
240 pg=INT(RND(1)*100)+1:IF ha(pg)=1 THE
   N 240                                     [BAA0]
242 IF t1=1 THEN IF prg(pg)>og OR prg(pg
   )<ug THEN 240                             [3E18]
250 f(t1,t)=pg                               [8A8E]
255 ha(pg)=1                                 [1236]
260 NEXT                                     [64EE]
270 NEXT                                     [60F0]
271 FOR t=1 TO 10                            [0FCE]
272 FOR t1=1 TO 10                           [3D32]
273 pr(t,t1)=49                              [DBBC]
274 NEXT                                     [5CF8]
275 NEXT                                     [5BFA]
280 CLS                                      [8D38]
290 PRINT:PRINT"Sie besitzen Rechte an f
   olgenden Programmen :"                    [C68C]
300 PRINT                                    [6180]
310 FOR t=1 TO 10                            [25C2]
320 PRINT SPC(25);:PRINT USING "\<20>\";
   prg$(f(1,t)),:PRINT prg(f(1,t))           [B9F4]
330 IF t<>10 THEN PRINT SPC(25);:FOR t2=
   1 TO 29:PRINT CHR$(154);:NEXT             [F8A6]
335 PRINT                                    [5E90]
340 NEXT                                     [77EC]
350 GOSUB 59000                              [3CAC]
400 REM ***  hauptmenue  ***                 [BB60]
405 FOR t=1 TO 10:FOR t1=1 TO 10:w(t,t1)
   =0:wsum(t,t1)=0:NEXT:NEXT                 [875A]
410 CLS                                      [882E]
415 PRINT:PRINT:PRINT"<2>Hauptmenue :"       [889E]
417 PRINT"<2>-------------"                  [59D2]
420 PRINT:PRINT"<2>[ w ]<2>:<2>Werbung f
   uer eigene Programme"                     [7474]
430 PRINT:PRINT"<2>[ n ]<2>:<2>Neuentwic
   klung eigener Programme"                  [7D96]
435 PRINT:PRINT"<2>[ p ]<2>:<2>Preise de
   r Programme neu festsetzen"               [112C]
440 PRINT:PRINT"<2>[ s ]<2>:<2>Statistik
   "                                         [8FEA]
450 PRINT:PRINT"<2>[ h ]<2>:<2>Highscore
   s anzeigen"                               [8956]
```

**Listing 2. Erobern Sie den Software-Markt**

```
460 PRINT:PRINT"<2>[ a ]<2>:<2>Spielstan
    d abspeichern"                          [AEC0]
470 PRINT:PRINT"<2>[ l ]<2>:<2>Spielstan
    d laden"                                [6CD8]
480 PRINT:PRINT"<2>[ SPACE ]<2>:<2>weite
    r im Spielverlauf"                      [B194]
490 LOCATE 20,24 : PRINT "Bitte waehlen
    Sie eine der Optionen : [<3>]";CHR$(
    8);CHR$(8);CHR$(8)                      [2D34]
500 a$=INKEY$:IF a$="" THEN 500             [D01C]
510 IF a$="w" THEN GOSUB 1500:GOTO 410      [D7BE]
520 IF a$="n" THEN GOSUB 3000:GOTO 410      [54A8]
525 IF a$="p" THEN GOSUB 4500:GOTO 410      [D2C2]
530 IF a$="s" THEN GOSUB 5000:GOTO 410      [DCB8]
540 IF a$="h" THEN GOSUB 7000:GOTO 410      [3CA8]
550 IF a$="a" THEN GOSUB 8000:GOTO 410      [D69E]
560 IF a$="l" THEN GOSUB 9000:GOTO 410      [50B8]
570 IF a$=" " THEN 590                      [3D3C]
580 GOTO 500                                [6C56]
585 REM *** Hauptschleife ***               [B58E]
590 CLS                                     [8E40]
595 IF woche=52 THEN 10000 : REM *** Sch
    lussuebersicht ***                      [75B2]
600 woche=woche + 1:flag=1:FOR t=1 TO 10
    0:ind(t)=ind(t)-0.01:NEXT t             [7408]
601 FOR w=1 TO 10:flag6=0:ez(w)=ez(w)-1:
    IF ez(w)=0 THEN wq=w:flag6=1::GOSUB
    57000                                   [B322]
602 NEXT w                                  [551C]
605 IF woche <> 1 THEN GOSUB 57000          [C4EC]
610 LOCATE 1,3:PRINT"Woche : ";woche;"<2
    0>Firma : ";f$(flag)                    [C406]
630 LOCATE 1,5:PRINT"Titel :<16>verkauft
    e Exemplare :<7>Gesamtverkauf bis je
    tzt :"                                  [2964]
640 GOSUB 58500                             [38B8]
645 IF flag<>1 THEN 720                     [4876]
650 FOR q=1 TO 10                           [95CA]
660 FOR q1=1 TO 10                          [A02E]
670 r=INT(RND(1)*2000)+1                    [2AAC]
680 r=r/100                                 [F33E]
690 vk(q,q1)=INT((prg(f(q,q1))*(200+w(q,
    q1)-pr(q,q1)-r))*ind(f(q,q1)))          [0A66]
695 gvk(q,q1)=gvk(q,q1)+vk(q,q1)            [6B9A]
700 NEXT q1                                 [D570]
710 NEXT q                                  [1A10]
720 FOR t=1 TO 10                           [28CC]
730 PRINT USING"\<29>\";prg$(f(flag,t));
    :PRINT USING "####";vk(flag,t);:PRIN
    T SPC(24);:PRINT USING "######";gvk(
    flag,t)                                 [14BA]
740 NEXT t:IF flag <>1 THEN 750             [75CE]
741 hsfl = 0 : FOR t=1 TO 10 : sum(t)=0
    : NEXT t                                [CB16]
742 FOR t=1 TO 10                           [26D4]
743 FOR t1=1 TO 10                          [8338]
744 sum(t)=sum(t)+vk(t,t1)                  [A6A4]
745 NEXT t1                                 [BA88]
746 NEXT t                                  [3028]
747 FOR w=1 TO 10                           [26E4]
748 IF sum(w)>hsf(swg) THEN hsf(swg)=sum
    (w) : hsff(swg)=w : hsfl=1 : h1$(swg
    )=f$(w)                                 [A8A0]
749 NEXT w                                  [0D34]
750 GOSUB 58500                             [5DBC]
751 svk=0:sgvk=0                            [A2C0]
752 FOR t=1 TO 10                           [09D6]
753 svk=svk+vk(flag,t)                      [FB76]
754 sgvk=sgvk+gvk(flag,t)                   [FBE2]
755 NEXT t                                  [2B28]
760 PRINT"Summe :";SPC(22);svk;SPC(23);s
    gvk                                     [7142]
770 PRINT:PRINT:PRINT"<2>[ a ]<6>:<2>Ver
    kaeufe der anderen Firmen anzeigen"     [7CB0]
780         PRINT"<2>[ SPACE ]<2>:<2>weite
    r im Spielverlauf"                      [4B8C]
790 PRINT:PRINT"<18>Bitte waehlen Sie ei
    ne der Optionen :<2>[<3>]";CHR$(8);C
    HR$(8);CHR$(8);                         [EF32]
800 a$=INKEY$:IF a$="" THEN 800             [7028]
810 IF a$="a" THEN GOSUB 58000:CLS:GOTO
    610                                     [8B42]
820 IF a$=" " THEN 840                      [2B34]
830 GOTO 800                                [DD58]
840 CLS                                     [C13C]
845 GOSUB 58500                             [42C6]
850 LOCATE 1,2:PRINT"Woche : ";woche;"<8
    >Die Hitliste der 18 bestverkauften
    Programme dieser Woche :"               [7454]
860 GOSUB 58500                             [3AC0]
870 PRINT"<4>Firma :<15>Titel :<16>Anzah
    l :<8>Punkte :"                         [BD3C]
880 GOSUB 58500                             [60C4]
885 f0=0                                    [8290]
890 FOR t=1 TO 10                           [31DC]
900 FOR t1=1 TO 10                          [DC2E]
910 IF vk(t,t1)>2300 THEN f0=f0+1:p(f0)=
    vk(t,t1):p1(f0)=t:p2(f0)=t1             [34E8]
920 NEXT t1                                 [E07E]
930 NEXT t                                  [561E]
940 a=f0                                    [FAE2]
945 p(0)=10000                              [80C2]
950 FOR x=2 TO a                            [A7E0]
960 IF p(x)<=p(x-1) THEN 1010               [51D0]
970 x0=p(x):x1=p1(x):x2=p2(x):FOR y=x-1
    TO 1 STEP -1                            [F15E]
980 p(y+1)=p(y):p1(y+1)=p1(y):p2(y+1)=p2
    (y)                                     [3544]
990 IF x0>=p(y-1) THEN 1000                 [8CB8]
995 p(y)=x0:p1(y)=x1:p2(y)=x2:GOTO 1010     [29A6]
1000 NEXT y                                 [A572]
1010 NEXT x                                 [B972]
1020 FOR t=1 TO 18                          [0330]
1021 IF t=1 THEN pkt=30:GOTO 1030           [BF54]
1022 IF t=2 THEN pkt=25:GOTO 1030           [7360]
1023 IF t=3 THEN pkt=20:GOTO 1030           [A45A]
1025 pkt=19-t                               [54FE]
1030 PRINT USING "##";t;:PRINT".";" ";:P
     RINT USING "\<20>\";f$(p1(t));:PRIN
     T USING "\<21>\";prg$(f(p1(t),p2(t)
     ));:PRINT p(t);SPC(11);:PRINT USING
      "##";pkt;                             [E924]
1035 IF p1(t)=1 THEN PRINT"<6><<<<<":ELS
     E PRINT                                [0050]
1037 pk(p1(t))=pk(p1(t))+pkt                [798C]
1040 NEXT t                                 [AF70]
1041 IF p(1)>hsw(swg) THEN hsw(swg)=p(1)
     :hswf(swg)=p1(1):hswt(swg)=f(p1(1),
     p2(1)):hsfl=1:h2$(swg)=f$(p1(1))       [3AE4]
1050 GOSUB 59000                            [DD08]
1055 IF hsfl = 1 THEN GOSUB 7000:GOSUB 5
     9800                                   [0A56]
1060 CLS                                    [1392]
1070 GOSUB 58500                            [5A14]
1080 PRINT"Woche : ";woche;"<14>Der aktu
     elle Stand im Softwarecup :"           [81D2]
1090 GOSUB 58500                            [E018]
1095 PRINT"<4>Firma :<20>Punkte :"          [13B4]
1096 GOSUB 58500                            [E824]
1100 FOR t=1 TO 10                          [A71E]
1110 po(t)=pk(t):po1(t)=t                   [B048]
1120 NEXT t                                 [A06E]
1130 a=10                                   [83C8]
1140 g=a-1:FOR x=a-1 TO 1 STEP -1           [35D8]
1150 d=0:FOR y=1 TO g                       [0E54]
1160 IF po(y)>=po(y+1) THEN 1180            [DBF2]
1170 f=y:s=po(y):s1=po1(y):po(y)=po(y+1)
     :po1(y)=po1(y+1):po(y+1)=s:po1(y+1)
     =s1                                    [F0F2]
1180 NEXT y                                 [9284]
1190 g=f:IF f=0 THEN 1210                   [F2C8]
1200 NEXT x                                 [C174]
1210 FOR t=1 TO 10                          [5622]
1220 PRINT USING"##";t;:PRINT".";" ";:PR
     INT USING"\<27>\";f$(po1(t));:PRINT
      USING "#####";po(t);                  [A850]
1225 IF po1(t)=1 THEN PRINT"<5><<<<<<<<<
     <":ELSE PRINT                          [5648]
1230 NEXT t                                 [A372]
1240 GOSUB 59000                            [E80A]
1245 flag=1                                 [F8E8]
1250 CLS                                    [0494]
1255 knt=0:kntg=0:knt2=0:kntg2=0            [066A]
1260 GOSUB 58500                            [0B16]
1270 PRINT"Woche : ";woche;"<10>Firma :
     ";:PRINT USING "\<19>\";f$(flag);:P
     RINT "<3>Aktueller Kontostand :"       [710C]
1280 GOSUB 58500                            [651A]
1290 PRINT"In den TOP 18 haben sich folg
     ende Programme plaziert :"             [D1BE]
1300 PRINT                                  [B1E2]
1305 PRINT"<4>Titel :<20>Menge :<8>Praem
     ie :<8>Gewinn :"                       [FC76]
1307 GOSUB 58500                            [0B1A]
1310 FOR t=1 TO 18:IF flag <> 1 THEN 132
     0                                      [C77A]
1311 IF t=1 THEN konto(p1(t))=konto(p1(t
     ))+300000+(vk(p1(t),p2(t))*(pr(p1(t
     ),p2(t))-29)):GOTO 1320                [72E8]
1312 IF t=2 THEN konto(p1(t))=konto(p1(t
     ))+250000+(vk(p1(t),p2(t))*(pr(p1(t
     ),p2(t))-29)):GOTO 1320                [DAF4]
1313 IF t=3 THEN konto(p1(t))=konto(p1(t
     ))+200000+(vk(p1(t),p2(t))*(pr(p1(t
     ),p2(t))-29)):GOTO 1320                [DEEE]
1314 konto(p1(t))=konto(p1(t))+((19-t)*1
     0000)+(vk(p1(t),p2(t))*(pr(p1(t),p2
     (t))-29))                              [6586]
1320 IF p1(t)=flag THEN PRINT USING "##"
     ;t;:PRINT". ";:PRINT USING "\<20>\"
     ;prg$(f(p1(t),p2(t)));:PRINT "<5>";
     vk(p1(t),p2(t));:ELSE GOTO 1390        [AF3C]
1330 IF t=1 THEN knt=300000:GOTO 1370       [6FE4]
1340 IF t=2 THEN knt=250000:GOTO 1370       [A8F0]
1350 IF t=3 THEN knt=200000:GOTO 1370       [20EA]
1360 knt = (19-t)*10000                     [1E56]
```

**Listing 2. Erobern Sie den Software-Markt (Fortsetzung)**

```
1370 PRINT SPC(9);:PRINT USING "######";
     knt;                                  [1C52]
1380 knt2 = vk(p1(t),p2(t)) * (pr(p1(t),
     p2(t)) - 29)                          [8BA4]
1385 PRINT SPC(11);:PRINT USING "######"
     ;knt2                                 [C69E]
1387 kntg=kntg+knt                         [E020]
1388 kntg2=kntg2+knt2                      [CA4E]
1390.NEXT t                                [B080]
1400 PRINT SPC(44);"---------------------
     ---------"                            [4438]
1410 PRINT SPC(54);:PRINT USING "#######
     ";kntg+kntg2                          [1668]
1415 PRINT                                 [70F0]
1416 PRINT"Momentaner Kontostand : ";kon
     to(flag)                              [25C6]
1420 LOCATE 1,20                           [F0FC]
1430 PRINT"[<2>a<2>]<3>:<3>Kontostaende
     anderer Firmen anzeigen "             [E59C]
1440 PRINT"[SPACE]<3>:<3>weiter im Spiel
     verlauf"                              [C5E0]
1450 PRINT:PRINT"<19>Bitte waehlen Sie e
     ine der Optionen : [<3>]";CHR$(8);C
     HR$(8);CHR$(8);                       [2B86]
1460 a$=INKEY$:IF a$="" THEN 1460          [48F4]
1470 IF a$=" " THEN 405                    [9B92]
1480 GOSUB 58000                           [2D14]
1490 GOTO 1250                             [2A1E]
1495 REM *** Werbung fuer eigene Program
     me ***                                [606C]
1500 CLS                                   [0990]
1510 GOSUB 58500                           [EE12]
1520 PRINT"<25>Werbung fuer eigene Progr
     amme"                                 [36B8]
1530 GOSUB 58500                           [E816]
1540 PRINT"<4>Titel :<22>Werbesumme (max
     . 10000) dieser Woche :"              [11EA]
1550 GOSUB 58500                           [E21A]
1560 FOR t=1 TO 10                         [9D32]
1570 PRINT"[";t-1;"] ";:PRINT USING "\<2
     8>\";prg$(f(1,t));:PRINT USING "###
     ##";wsum(1,t)                         [5196]
1580 NEXT                                  [2E5A]
1590 PRINT:PRINT"[SPACE]<3>:<3>Rueckkehr
      zum Hauptmenue"                      [2702]
1600 LOCATE 1,24                           [7704]
1610 PRINT"<20>Bitte waehlen Sie ein Pro
     gramm an : [<3>]";CHR$(8);CHR$(8);C
     HR$(8);                               [F984]
1620 a$=INKEY$:IF a$="" THEN 1620          [DAEC]
1630 IF a$=" " THEN RETURN                 [B61C]
1640 IF VAL(a$)<0 OR VAL(a$)>9 THEN 1620
                                           [C710]
1650 LOCATE 38,VAL(a$)+6                   [3BEA]
1660 INPUT wsum(1,VAL(a$)+1):IF wsum(1,V
     AL(a$)+1)>10000 THEN wsum(1,VAL(a$)
     +1)=10000                             [CAFE]
1661 IF konto(1)<wsum(1,VAL(a$)+1) THEN
     wsum(1,VAL(a$)+1)=0:GOTO 1600         [A4B4]
1662 konto(1)=konto(1)-wsum(1,VAL(a$)+1)
                                           [7284]
1663 FOR t=1 TO 10:r1(t)=INT(RND(1)*25)+
     1:NEXT t                              [0E30]
1665 w(1,VAL(a$)+1)=INT(wsum(1,VAL(a$)+1
     )/(400-r1(1)))                        [1134]
1670 FOR t=2 TO 10                         [7038]
1675 IF konto(t)<wsum(1,VAL(a$)+1) THEN
     1690                                  [0838]
1680 wsum(t,VAL(a$)+1)=wsum(1,VAL(a$)+1)
                                           [8D1A]
1685 w(t,VAL(a$)+1)=INT(wsum(t,VAL(a$)+1
     )/(400-r1(t)))                        [F9CA]
1687 konto(t)=konto(t)-wsum(t,VAL(a$)+1)
                                           [CA24]
1690 NEXT t                                [CF86]
1700 GOTO 1600                             [A210]
2995 REM *** Neuentwicklung eigener Prog
     ramme ***                             [D0AA]
3000 CLS                                   [178A]
3010 GOSUB 58500                           [5E0C]
3020 PRINT"<18>Neuentwicklung eigener Pr
     ogramme :"                            [B0D8]
3030 GOSUB 58500                           [E810]
3040 FOR t=0 TO 9                          [0ED6]
3050 PRINT"[<2>";t;"<2>]<3>:<3>";prg$(f(
     1,t+1))                               [A2EC]
3060 NEXT t                                [C378]
3070 PRINT:PRINT"[SPACE]<3>:<3>Rueckkehr
      zum Hauptmenue"                      [B8F8]
3080 LOCATE 1,24                           [210C]
3090 PRINT"<18>Bitte waehlen Sie ein Pro
     gramm an : [<3>]";CHR$(8);CHR$(8);C
     HR$(8);                               [280C]
3100 a$=INKEY$:IF a$="" THEN 3100          [3DD8]
3110 IF a$=" " THEN RETURN                 [3A12]
3120 IF VAL(a$)<0 OR VAL(a$)>9 THEN 3100
                                           [F4FC]
3125 IF ez(VAL(a$)+1)>0 THEN LOCATE 45,4
     +VAL(a$) : PRINT">Programm ist in E
     ntwicklung !<":FOR t=1 TO 2000:NEXT
      t:GOTO 3000                          [6F92]
3130 CLS                                   [1292]
3140 GOSUB 58500                           [E714]
3150 PRINT"<21>Neuentwicklung eigener Pr
     ogramme :"                            [CBA0]
3160 GOSUB 58500                           [5D18]
3170 PRINT"Titel : ";prg$(f(1,VAL(a$)+1)
     )                                     [29B4]
3180 PRINT"Momentane Qualitaet : ";prg(f
     (1,VAL(a$)+1))                        [1246]
3190 PRINT"Momentaner<4>Index : ";ind(f(
     1,VAL(a$)+1))                         [926C]
3200 GOSUB 58500                           [0B0E]
3210 INPUT"Neue Qualitaet : ";nqua         [9EF6]
3215 IF nqua<prg(f(1,VAL(a$)+1)) OR nqua
     >25 THEN 3210                         [D77C]
3220 PRINT"Neuer<4>Index :<2>1"            [F3F4]
3230 INPUT"Entwicklungszeit in Wochen :
     ";ewz                                 [70F8]
3240 IF ewz<1 OR ewz>10 THEN 3230          [11D0]
3250 GOSUB 58500                           [E618]
3255 kostn=INT((11-ewz)*((nqua-prg(f(1,V
     AL(a$)+1)))*100000) + (((1-ind(f(1,
     VAL(a$)+1)))*100000)*(11-ewz)))       [C04A]
3260 PRINT"Kosten : ";kostn                [8828]
3270 IF konto(1)<kostn THEN PRINT "Entwi
     cklung wegen Geldmangel nicht moegl
     ich !":GOTO 3400                      [543A]
3280 PRINT"Programm in dieser Form entwi
     ckeln (j/n) ? : [<5>]";CHR$(8);CHR$
     (8);CHR$(8);CHR$(8);CHR$(8);          [21B6]
3290 b$=INKEY$:IF b$="" THEN 3290          [D104]
3300 IF b$="n" THEN 3400                   [DA80]
3310 IF b$<>"j" THEN 3290                  [B202]
3320 PRINT"OK!"                            [DCE8]
3330 konto(1)=konto(1)-kostn               [93B8]
3340 ez(VAL(a$)+1)=ewz                     [3184]
3350 nq(VAL(a$)+1)=nqua                    [D544]
3360 kt(VAL(a$)+1)=kostn                   [743A]
3400 GOSUB 59000                           [E60A]
3410 RETURN                                [E390]
4495 REM *** Preise neu festsetzen ***     [C6E2]
4500 CLS                                   [1A96]
4510 GOSUB 58500                           [6118]
4520 PRINT"<21>Preise der Programme neu
     festsetzen"                           [5B78]
4530 GOSUB 58500                           [EB1C]
4540 PRINT "<14>Titel :<21>Preis :"        [6BEE]
4545 GOSUB 58500                           [EC28]
4550 FOR t=0 TO 9                          [2DE4]
4560 LOCATE 1,(t+1)+5                      [824A]
4570 PRINT"[ ";t;" ]<3>:<3>";:PRINT USIN
     G "\<27>\";prg$(f(1,t+1));:PRINT pr
     (1,t+1)                               [2AC8]
4580 NEXT                                  [F660]
4585 PRINT                                 [D906]
4590 PRINT"[SPACE]<3>:<3>Rueckkehr zum H
     auptmenue"                            [3F7A]
4600 LOCATE 1,24                           [CC0A]
4610 PRINT "<20>Bitte waehlen Sie Progra
     mm an : [<3>]";CHR$(8);CHR$(8);CHR$
     (8);                                  [6B12]
4620 a$=INKEY$:IF a$="" THEN 4620          [3DF8]
4630 IF a$=" " THEN RETURN                 [3B22]
4640 IF VAL(a$)<0 OR VAL(a$)>9 THEN 4620
                                           [8B1C]
4650 LOCATE 43,VAL(a$)+6                   [07E8]
4660 INPUT pr(1,VAL(a$)+1)                 [4F8A]
4665 IF pr(1,VAL(a$)+1)<29 OR pr(1,VAL(a
     $)+1)>69 THEN 4650                    [A3C2]
4670 FOR t=2 TO 9                          [CCEE]
4680 pr(t,VAL(a$)+1)=pr(1,VAL(a$)+1)       [2E78]
4690 NEXT t                                [DB8C]
4700 GOTO 4600                             [681C]
4995 REM *** Statistik ***                 [CDF6]
5000 CLS                                   [298E]
5010 GOSUB 58500                           [EC10]
5020 PRINT"<33>Statistik"                  [8230]
5030 GOSUB 58500                           [6214]
5035 PRINT:PRINT                           [5B82]
5040 PRINT"Untermenue Statistik :"         [1038]
5050 PRINT"----------------------"         [0E32]
5060 PRINT                                 [A5F0]
5070 PRINT"[<2>1<2>]<2>:<2>Bisherige Ges
     amtverkaeufe anderer Firmen anzeige
     n"                                    [5948]
5080 PRINT"[<2>2<2>]<2>:<2>Bisherige Ges
     amtverkaeufe aller Programme anzeig
     en"                                   [143C]
5090 PRINT"[<2>3<2>]<2>:<2>Programmverte
     ilung auf die verschiedenen Firmen
     anzeigen"                             [A722]
5100 PRINT"[<2>4<2>]<2>:<2>In der Neuent
     wicklung befindliche Programme anze
     igen"                                 [9254]
5110 PRINT"[<2>5<2>]<2>:<2>Kontostaende
     aller Firmen anzeigen"                [E6E0]
5120 PRINT"[<2>6<2>]<2>:<2>Aktuellen Sta
```







```
     nd im Softwarecup anzeigen"          [9874]
5130 PRINT"[SPACE]<2>:<2>Rueckkehr zum H
     auptmenue"                           [0CE8]
5140 LOCATE 20,24                         [4E6C]
5150 PRINT"Bitte waehlen Sie eine der Op
     tionen : [<3>]";CHR$(8);CHR$(8);CHR
     $(8);                                [233A]
5160 a$=INKEY$:IF a$="" THEN 5160         [3DF8]
5170 IF a$=" " THEN RETURN                [3E22]
5180 IF VAL(a$)<1 OR VAL(a$)>6 THEN 5160
                                          [4018]
5190 ON VAL(a$) GOSUB 5200,5500,5800,610
     0,6400,6700 : GOTO 5000              [09AA]
5199 REM ** Gesamtverkaeufe / Firmen anz
     eigen **                             [DC88]
5200 CLS                                  [2192]
5210 GOSUB 58500                          [F014]
5220 PRINT"Woche : ";woche;"<14>Statisti
     k<2>==><3>Gesamtverkaeufe / Firmen"
                                          [6B82]
5230 GOSUB 58500                          [EA18]
5240 PRINT"<4>Firma :<28>Gesamtverkauf b
     is jetzt :"                          [11C6]
5250 GOSUB 58500                          [5C1C]
5255 FOR t=1 TO 10 : gksum(t)=0 : NEXT t
                                          [907C]
5260 FOR t=1 TO 10                        [6334]
5270 FOR t1=1 TO 10                       [C798]
5280 gksum(t)=gksum(t)+gvk(t,t1)          [461A]
5290 NEXT t1                              [EDE8]
5300 gkflag(t)=t                          [7694]
5310 NEXT t                               [B878]
5320 a=10                                 [9DD2]
5330 g=a-1:FOR x=a-1 TO 1 STEP -1         [1BE2]
5340 d=0:FOR y=1 TO g                     [685E]
5350 IF gksum(y)>=gksum(y+1) THEN 5370    [1A26]
5360 f=y:s=gksum(y):s1=gkflag(y):gksum(y
     )=gksum(y+1):gkflag(y)=gkflag(y+1):
     gksum(y+1)=s:gkflag(y+1)=s1          [6D1C]
5370 NEXT y                               [B68E]
5380 g=f:IF f=0 THEN 5400                 [7BDC]
5390 NEXT x                               [8090]
5400 FOR t=1 TO 10                        [592C]
5410 PRINT USING "##";t;:PRINT". ";:PRIN
     T USING "\<43>\";f$(gkflag(t));:PRI
     NT USING "#######";gksum(t);         [38B0]
5420 IF gkflag(t)=1 THEN PRINT "<6><<<<<
     <<<<<":ELSE PRINT                    [C482]
5430 NEXT t                               [CD7E]
5440 GOSUB 59000                          [F016]
5450 RETURN                               [CB9C]
5499 REM ** Gesamtverkaeufe / Programme
     anzeigen **                          [A320]
5500 CLS                                  [0598]
5505 z=0                                  [08AC]
5510 GOSUB 58500                          [E21A]
5520 PRINT"Woche : ";woche;"<12>Statisti
     k<2>==><3>Gesamtverkaeufe / Program
     me"                                  [BF9A]
5530 GOSUB 58500                          [0C1E]
5540 grenze = 2500*woche                  [3CFA]
5550 FOR t=1 TO 10                        [8238]
5560 FOR t1=1 TO 10                       [D09C]
5570 IF gvk(t,t1)>grenze THEN z=z+1:gv(z
     )=gvk(t,t1):gv1(z)=t:gv2(z)=t1       [06F4]
5580 NEXT t1                              [5EEC]
5590 NEXT t                               [DA8C]
5600 a=z:gv(0)=10000000                   [C6B2]
5610 FOR x=2 TO a                         [AF3C]
5620 IF gv(x)<=gv(x-1) THEN 5670          [7000]
5630 x0=gv(x):x1=gv1(x):x2=gv2(x):FOR y=
     x-1 TO 1 STEP -1                     [5648]
5640 gv(y+1)=gv(y):gv1(y+1)=gv1(y):gv2(y
     +1)=gv2(y)                           [C9BC]
5650 IF x0>=gv(y-1) THEN 5660             [B40E]
5655 gv(y)=x0:gv1(y)=x1:gv2(y)=x2:GOTO 5
     670                                  [C4B0]
5660 NEXT y                               [AB92]
5670 NEXT x                               [BF92]
5672 PRINT"<4>Firma :<15>Titel :<16>Anza
     hl :"                                [DB04]
5673 GOSUB 58500                          [E82E]
5674 FOR t=1 TO 18                        [E856]
5675 PRINT USING "##";t;:PRINT".";" ";:P
     RINT USING "\<20>\";f$(gv1(t));:PRI
     NT USING "\<22>\";prg$(f(gv1(t),gv2
     (t)));:PRINT USING "######";gv(t);   [E580]
5677 IF gv1(t)=1 THEN PRINT "<5><<<<<<<<
     <" : ELSE PRINT                      [4CAA]
5680 NEXT t                               [BD8C]
5685 GOSUB 59000                          [E82C]
5690 RETURN                               [C1A8]
5799 REM ** Programmverteilung anzeigen
     **                                   [E9CA]
5800 CLS                                  [999E]
5810 GOSUB 58500                          [0C20]
5820 PRINT"<22>Statistik<2>==><3>Program
     mverteilung"                         [DE44]
5830 GOSUB 58500                          [6224]
5840 PRINT                                [DFFC]
5850 FOR t=0 TO 9                         [2EEC]
5860 PRINT "[<2>";t;"<2>]<3>:<3>";f$(t+1
     )                                    [AD54]
5870 NEXT t                               [898E]
5880 PRINT                                [A404]
5890 PRINT"[ SPACE ]<3>:<3>Rueckkehr zum
      Untermenue Statistik"               [D2DA]
5900 LOCATE 20,24                         [5674]
5910 PRINT"Bitte waehlen Sie ein Program
     m an : [<3>]";CHR$(8);CHR$(8);CHR$(
     8);                                  [0892]
5920 a$=INKEY$:IF a$="" THEN 5920         [1208]
5930 IF a$=" " THEN RETURN                [522A]
5940 IF VAL(a$)<0 OR VAL(a$)>9 THEN 5920
                                          [D12C]
5950 CLS                                  [26AA]
5960 GOSUB 58500                          [EF2C]
5970 PRINT"<5>Statistik<2>==><3>Programm
     verteilung<3>/<3>Firma : ";f$(VAL(a
     $)+1)                                [9816]
5980 GOSUB 58500                          [6930]
5990 PRINT"Titel :<20>Qualitaet :<14>Ver
     kauf bis jetzt :"                    [8E8A]
6000 GOSUB 58500                          [F410]
6010 FOR t=1 TO 10                        [9928]
6020 PRINT USING "\<27>\";prg$(f(VAL(a$)
     +1,t));:PRINT USING "##.##";prg(f(V
     AL(a$)+1,t));:PRINT "<22>";:PRINT U
     SING "######";gvk(VAL(a$)+1,t)       [BF20]
6030 NEXT t                               [BD78]
6040 GOSUB 59000                          [5910]
6050 RETURN                               [0396]
6099 REM ** Neuentwicklungen anzeigen **
                                          [48F6]
6100 CLS                                  [3492]
6110 GOSUB 58500                          [EF14]
6120 PRINT"Woche : ";woche;"<14>Statisti
     k<2>==><3>Neuentwicklungen"          [94F0]
6130 GOSUB 58500                          [F518]
6132 PRINT"Titel :<19>Wochen :<8>n.Qual.
      :<9>Kosten :"                       [24B8]
6134 GOSUB 58500                          [5D20]
6140 FOR t=1 TO 10                        [6830]
6150 IF ez(t) > 0 THEN PRINT USING "\<26
     >\";prg$(f(1,t));:PRINT USING "##";
     ez(t);:PRINT"<14>";:PRINT USING "##
     .##";nq(t);:PRINT"<11>";:PRINT USIN
     G "#######";kt(t)                    [C19C]
6160 NEXT t                               [C680]
6170 GOSUB 59000                          [5A18]
6180 RETURN                               [A89E]
6399 REM ** Kontostaende aller Firmen an
     zeigen **                            [67F0]
6400 CLS                                  [4098]
6410 GOSUB 58500                          [751A]
6420 PRINT"Woche : ";woche;"<17>Statisti
     k<2>==><3>Kontostaende"              [1E48]
6430 GOSUB 58500                          [2F1E]
6440 PRINT"<4>Firma :<24>Kontostand :"    [FF0E]
6450 GOSUB 58500                          [E922]
6460 FOR t=1 TO 10                        [A43A]
6470 ko(t)=konto(t):ko1(t)=t              [0FF0]
6480 NEXT t                               [DB8A]
6490 a=10                                 [A6E4]
6500 g=a-1:FOR x=a-1 TO 1 STEP -1         [41E2]
6510 d=0:FOR y=1 TO g                     [0A5E]
6520 IF ko(y)>=ko(y+1) THEN 6540          [DCF2]
6530 f=y:s=ko(y):s1=ko1(y):ko(y)=ko(y+1)
     :ko1(y)=ko1(y+1):ko(y+1)=s:ko1(y+1)
     =s1                                  [1BAC]
6540 NEXT y                               [C88E]
6550 g=f:IF f=0 THEN 6570                 [D9EE]
6560 NEXT x                               [D290]
6570 FOR t=1 TO 10                        [B33E]
6580 PRINT USING "##";t;:PRINT ". ";:PRI
     NT USING "\<30>\";f$(ko1(t));:PRINT
      USING "##########";ko(t);           [3FF2]
6590 IF ko1(t)=1 THEN PRINT "<5><<<<<<<<
     <<":ELSE PRINT                       [AC92]
6600 NEXT t                               [A57E]
6610 GOSUB 59000                          [3816]
6620 RETURN                               [CF9C]
6699 REM ** Aktuellen Stand im Softwarec
     up anzeigen **                       [F486]
6700 CLS                                  [349E]
6710 GOSUB 58500                          [1320]
6720 PRINT"Woche : ";woche;"<13>Statisti
     k<2>==><3>Stand des Softwarecups"    [0468]
6730 GOSUB 58500                          [2D24]
6740 PRINT"<4>Firma :<20>Punkte :"        [1DB8]
6750 GOSUB 58500                          [1F28]
6760 FOR t=1 TO 10                        [A640]
6770 PRINT USING"##";t;:PRINT".";" ";:PR
     INT USING"\<27>\";f$(po1(t));:PRINT
      USING "#####";po(t);                [656E]
6780 IF po1(t)=1 THEN PRINT"<5><<<<<<<<<
```

**Listing 2. Erobern Sie den Software-Markt (Fortsetzung)**

```
     <":ELSE PRINT                                [A85E]
6790 NEXT t                                       [EE92]
6800 GOSUB 59000                                  [2D18]
6810 RETURN                                       [8B9E]
6990 REM *** Highscores anzeigen ***              [0070]
7000 CLS                                          [2392]
7010 GOSUB 58500                                  [F214]
7020 PRINT"<13>HIGHSCORES<16>Schwierigke
     itsgrad : ";swg-1                            [3A20]
7030 GOSUB 58500                                  [EC18]
7032 IF hsfl=1 THEN sg=swg:GOTO 7040              [3FD6]
7035 LOCATE 60,2:INPUT sg:sg=sg+1:IF sg<
     1 OR sg>8 THEN 7035                          [EBEA]
7040 PRINT:PRINT" 1. woechentlicher Absa
     tz :"                                        [73AA]
7050   PRINT"------------------------------
     "                                            [D978]
7060 PRINT"Firma : ";:PRINT USING "\<20>
     \";h2$(sg);:PRINT"<2>Titel : ";:PRI
     NT USING "\<17>\";prg$(hswt(sg));:P
     RINT"<2>Menge : ";hsw(sg)                    [2806]
7070 PRINT:PRINT" 2. woechentlicher Absa
     tz einer Firma :"                            [B036]
7080        PRINT"-------------------------
     -----------------                            [98CC]
7090 PRINT"Firma : ";:PRINT USING "\<20>
     \";h1$(sg);:PRINT"<2>Menge : ";hsf(
     sg)                                          [3F48]
7100 PRINT:PRINT" 3. Gesamtpunktzahl ein
     es Jahres :"                                 [9DDC]
7110   PRINT"------------------------------
     -------"                                     [F5E8]
7120 PRINT"Firma : ";:PRINT USING "\<20>
     \";h3$(sg);:PRINT"<2>Punkte: ";hsp(
     sg)                                          [B82A]
7997 hsfl=0                                       [C134]
7998 GOSUB 59000 : RETURN                         [A6F2]
7999 REM *** Spielstand auf Diskette abs
     peichern ***                                 [C06C]
8000 CLS                                          [2A94]
8010 GOSUB 58500                                  [D316]
8020 PRINT"<21>Spielstand auf Diskette a
     bspeichern"                                  [C79E]
8030 GOSUB 58500                                  [E91A]
8040 PRINT:PRINT"Bisher existieren folge
     nde 'SOFT - CHEF' - Files :"                 [6E18]
8050 |DIR,"sc-*.dat"                              [4798]
8060 PRINT:PRINT:PRINT"Welchen Namen sol
     l das neue File erhalten (ohne 'sc-
     '!) : ";                                     [C22C]
8070 INPUT file$:file$=LEFT$(file$,5):fi
     le$=file$+".dat"                             [A870]
8080 filename$="sc-"+file$                        [7390]
8090 OPENOUT filename$                            [4740]
8100 WRITE #9,woche                               [4C64]
8105 WRITE #9,f$(1)                               [8B5A]
8107 WRITE #9,swg                                 [60E8]
8110 FOR t=1 TO 10                                [5F2E]
8120 FOR t1=1 TO 10                               [C492]
8130 WRITE #9,f(t,t1)                             [4936]
8132 WRITE #9,gvk(t,t1)                           [9DFE]
8134 WRITE #9,pr(t,t1)                            [AC36]
8140 NEXT t1                                      [CDE2]
8150 NEXT t                                       [8C82]
8210 FOR t=1 TO 10                                [9B30]
8220 WRITE #9,ez(t)                               [EF86]
8222 WRITE #9,nq(t)                               [148A]
8224 WRITE #9,kt(t)                               [258E]
8230 WRITE #9,pk(t)                               [2080]
8240 WRITE #9,konto(t)                            [5D22]
8250 NEXT t                                       [C784]
8260 FOR t=1 TO 100                               [9B9A]
8270 WRITE #9,ind(t)                              [BA48]
8280 WRITE #9,prg(t)                              [CB66]
8290 NEXT t                                       [B78C]
8295 CLOSEOUT                                     [5ACC]
8300 RETURN                                       [9A96]
8999 REM *** Laden des Spielstands von D
     iskette ***                                  [F8FA]
9000 CLS                                          [3596]
9010 GOSUB 58500                                  [F018]
9020 PRINT"<23>Laden des Spielstands von
      Diskette"                                   [B0AC]
9030 GOSUB 58500                                  [F61C]
9040 PRINT:PRINT"Folgende 'SOFT - CHEF'
     - Files existieren auf dieser Diske
     tte :"                                       [518A]
9050 PRINT:|DIR,"sc-*.dat"                        [CD28]
9060 PRINT:PRINT:PRINT"Name des Files, d
     as geladen werden soll (ohne 'sc-'!
     ) : ";                                       [7926]
9070 INPUT file$:file$=LEFT$(file$,5):fi
     le$=file$+".dat"                             [1172]
9080 filename$="sc-"+file$                        [4C92]
9090 OPENIN filename$                             [6A80]
9100 INPUT#9,woche                                [D630]
9105 INPUT#9,f$(1)                                [5226]
9107 INPUT #9,swg                                 [EEF4]
9110 FOR t=1 TO 10                                [9A30]
9120 FOR t1=1 TO 10                               [A494]
9130 INPUT #9,f(t,t1)                             [8542]
9132 INPUT #9,gvk(t,t1)                           [CA0A]
9134 INPUT #9,pr(t,t1)                            [CA42]
9140 NEXT t1                                      [2EE4]
9150 NEXT t                                       [C684]
9210 FOR t=1 TO 10                                [5E32]
9220 INPUT #9,ez(t)                               [1192]
9222 INPUT #9,nq(t)                               [FA96]
9224 INPUT #9,kt(t)                               [5B9A]
9230 INPUT #9,pk(t)                               [7A8C]
9240 INPUT #9,konto(t)                            [852E]
9250 NEXT t                                       [8B86]
9260 FOR t=1 TO 100                               [7D9C]
9270 INPUT #9,ind(t)                              [0654]
9280 INPUT #9,prg(t)                              [9F72]
9290 NEXT t                                       [DB8E]
9295 CLOSEIN                                      [710C]
9300 RETURN                                       [C898]
9995 REM *** Schlussuebersicht ***                [91A6]
10000 CLS                                         [B2E6]
10010 GOSUB 58500                                 [8B68]
10020 PRINT"Woche : ";woche;"<20>Schluss
      uebersicht"                                 [AA6E]
10030 GOSUB 58500                                 [A16C]
10040 PRINT:PRINT                                 [01D2]
10050 PRINT"Herzlichen Glueckwunsch ! Si
      e haben ein Jahr lang als Manager
      einer Software-"                            [B0DE]
10060 PRINT:PRINT"firma durchgehalten. Z
      um Abschluss werden Sie nun die En
      dstaende einiger Sta-"                      [AB94]
10070 PRINT:PRINT"tistiken sehen. Danach
       erhalten Sie fuer ihre Arbeit Pun
      kte, je nachdem wie gut"                    [B32E]
10080 PRINT:PRINT"oder wie schlecht Sie
      gearbeitet haben."                          [9DEE]
10090 GOSUB 59000                                 [6A70]
10100 GOSUB 5200                                  [6CF2]
10110 GOSUB 6400                                  [5DFA]
10120 GOSUB 6700                                  [3C02]
10130 FOR t=1 TO 10 : IF gkflag(t)=1 THE
      N put1=(11-t)*10+INT(gksum(t)/1000
      0)                                          [6B24]
10135 NEXT t                                      [45DA]
10140 FOR t=1 TO 10 : IF ko1(t)=1 THEN p
      ut2=(11-t)*10+INT(ko(t)/100000)             [992C]
10145 NEXT t                                      [31DC]
10150 FOR t=1 TO 10 : IF po1(t)=1 THEN p
      ut3=(11-t)*10+INT(po(t)/10)                 [D1C4]
10155 NEXT t                                      [3DDE]
10160 gput=put1+put2+put3                         [1918]
10170 CLS                                         [B3F6]
10180 GOSUB 58500                                 [6678]
10190 PRINT SPC(32);"Schlussabrechnung"           [B1CA]
10200 GOSUB 58500                                 [4E6A]
10210 PRINT:PRINT                                 [D6D0]
10220 PRINT"Sie erhalten folgende Punktz
      ahlen fuer Ihre Angstrengungen :"           [5098]
10230 PRINT:PRINT"Punkte fuer das Verkau
      fsergebnis Ihrer Programme .......
      ................";put1                      [1384]
10240 PRINT:PRINT"Punkte fuer Ihren aktu
      ellen Kontostand .................
      ................";put2                      [64BC]
10250 PRINT:PRINT"Punkte fuer Ihr Abschn
      eiden im Softwarecup .............
      ................";put3                      [FE90]
10260  PRINT"<70>-------"                         [190A]
10270 PRINT"Ihre Gesamtpunktzahl .......
      ..................................
      .........";gput                             [3CD8]
10280 PRINT:GOSUB 58500                           [0308]
10290 PRINT:PRINT"Highscore fuer Gesamtp
      unktzahl :"                                 [0BBC]
10300 PRINT:PRINT"Firma : ";h3$(swg);SPC
      (20);"Punkte : ";hsp(swg)                   [5F3A]
10310 IF gput>hsp(swg) THEN PRINT:PRINT"
      Sie haben einen neuen Highscore er
      reicht !":hsp(swg)=gput:h3$(swg)=f
      $(1):GOSUB 59800                            [9774]
10320 GOSUB 59000                                 [A768]
10330 CLS                                         [A3F2]
10340 LOCATE 20,12:PRINT"Wollen Sie noch
       einmal spielen (j/n) : ";:INPUT a
      $                                           [2850]
10350 IF a$="j" THEN RUN                          [BF38]
10355 CLEAR : CLS                                 [22C2]
10360 END                                         [D8E2]
56990 REM *** Neuentwicklungen der ander
      en Firmen ***                               [F6D8]
57000 CLS                                         [BBFC]
57005 flag5=0                                     [68BA]
57010 flag2=flag2+2:IF flag2=10 THEN fla
      g3=10:flag4=2:flag2=1:GOTO 57040            [2CF4]
57020 IF flag2=9 THEN flag3=9:flag4=10:f
      lag2=0:GOTO 57040                           [64DA]
57030 flag3=flag2:flag4=flag3+1                   [AFC6]
57040 GOSUB 58500                                 [A784]
```

```
57050 PRINT"<27>Neuentwicklungen dieser
      Woche :"                          [A708]
57060 GOSUB 58500                       [9188]
57065 IF flag6=1 THEN GOSUB 57700:GOTO 5
      7350                              [F87A]
57070 PRINT"Firma : ";f$(flag3)         [1F80]
57080 FOR t=1 TO 10                     [BFA2]
57090 in(t)=ind(f(flag3,t)):in2(t)=t    [0CCE]
57100 NEXT t                            [38E0]
57110 a=10                              [7C3A]
57120 FOR x=2 TO a                      [10A2]
57130 IF in(x)>=in(x-1) THEN 57190      [95BA]
57140 x0=in(x):x2=in2(x):FOR y=x-1 TO 1
      STEP -1                           [16A8]
57150 in(y+1)=in(y):in2(y+1)=in2(y)     [92EC]
57160 IF x0<=in(y-1) THEN 57180         [B2CC]
57170 in(y)=x0:in2(y)=x2:GOTO 57190     [5C6E]
57180 NEXT y                            [62FA]
57190 NEXT x                            [8EFA]
57200 PRINT"Titel : ";prg$(f(flag3,in2(1
      )))                               [5C3E]
57210 IF prg(f(flag3,in2(1))) <= 10 THEN
       stg=10 : GOTO 57250              [66DC]
57220 IF prg(f(flag3,in2(1))) <= 15 THEN
       stg= 8 : GOTO 57250              [57D6]
57230 IF prg(f(flag3,in2(1))) <= 20 THEN
       stg= 5 : GOTO 57250              [11CA]
57240 stg=25-prg(f(flag3,in2(1)))       [D22C]
57250 mstg=INT(konto(flag3)/100000)     [B7E0]
57260 IF mstg>=stg THEN wstg=stg:GOTO 57
      280                               [AA5E]
57270 wstg=mstg                         [EAC4]
57280 prg(f(flag3,in2(1)))=prg(f(flag3,i
      n2(1)))+wstg                      [A6FA]
57290 PRINT"Qualitaet der Neuentwicklung
       : ";prg(f(flag3,in2(1)))         [D4F4]
57300 kosten = INT(wstg*100000 + ((1-in(
      1))*100000))                      [0316]
57310 IF kosten > konto(flag3) THEN PRIN
      T:PRINT"Neuentwicklung wegen Geldm
      angel nicht beendet !":GOTO 57340 [F19C]
57320 PRINT"Kosten der Neuentwicklung<4>
      : ";kosten                        [5CBA]
57330 konto(flag3)=konto(flag3)-kosten:P
      RINT"Momentaner Kontostand : ";kon
      to(flag3)                         [892C]
57335 ind(f(flag3,in2(1))) = 1          [9238]
57340 IF flag5=0 THEN flag3=flag4 :flag5
      =1:GOSUB 58500:GOTO 57070         [AEBC]
57350 GOSUB 59000                       [5784]
57355 CLS                               [B716]
57360 RETURN                            [DD0A]
57700 PRINT"Firma : ";f$(1)             [6848]
57710 PRINT"Titel : ";prg$(f(1,wq))     [08CC]
57720 PRINT"Neue Qualitaet : ";nq(wq):pr
      g(f(1,wq))=nq(wq)                 [247E]
57730 ind(f(1,wq))=1                    [C438]
57740 PRINT"Kosten : ";kt(wq)           [E572]
57750 flag6=0                           [59CA]
57760 RETURN                            [E112]
57990 REM *** Firmen anwaehlen ***      [3F44]
58000 CLS                               [BAFE]
58020 FOR t=2 TO 10                     [EE9A]
58025 LOCATE 20,(t-1)*2+1               [64CA]
58030 PRINT"[ ";t-1;" ]<3>:<3>";f$(t):PR
      INT                               [B880]
58040 NEXT                              [CDC0]
58050 LOCATE 1,24:PRINT"<20>Bitte waehle
      n Sie eine der Firmen : [<3>]";CHR
      $(8);CHR$(8);CHR$(8);             [FE7C]
58060 a$=INKEY$:IF a$="" THEN 58060     [75D4]
58070 IF VAL(a$)<1 OR VAL(a$)>9 THEN 580
      60                                [17F8]
58080 flag=VAL(a$)+1                    [4A22]
58090 RETURN                            [180C]
58490 REM *** unterstreichen ***        [49BA]
58500 FOR t=1 TO 80:PRINT CHR$(154);:NEX
      T t                               [5DE2]
58510 RETURN                            [E706]
58990 REM *** auf SPACE warten ***      [6ED0]
59000 LOCATE 31,24:PRINT"< SPACE druecke
      n >";                             [2E92]
59010 a$=INKEY$:IF a$="" THEN 59010     [DDC4]
59020 IF a$=" " THEN RETURN             [7D88]
59030 GOTO 59010                        [1BF2]
59490 REM *** Wahl des Schwierigkeitsgra
      ds ***                            [8B92]
59500 CLS                               [620A]
59510 LOCATE 1,8                        [A522]
59520 PRINT "Welchen Schwierigkeitsgrad
      ( 0-7 ) wuenschen Sie : ";        [AF14]
59530 a$=INKEY$:IF a$="" THEN 59530     [67E0]
59540 IF VAL(a$)<0 OR VAL(a$)>7 THEN 595
      30                                [C8FE]
59550 swg=VAL(a$)+1:ug=20-1.5*swg:og=ug+
      10                                [71E2]
59560 RETURN                            [BF12]
59695 REM *** Highscores einlesen ***   [33E8]
59700 OPENIN "softchef.hsc"             [B920]
59710 FOR t=1 TO 8                      [7754]
59720 INPUT #9,hsf(t)                   [AECA]
59730 INPUT #9,hsff(t)                  [4498]
59740 INPUT #9,hsw(t)                   [E9F0]
59750 INPUT #9,hswf(t)                  [32BE]
59760 INPUT #9,hswt(t)                  [7EDC]
59762 INPUT #9,h1$(t)                   [CFCE]
59764 INPUT #9,h2$(t)                   [19D4]
59766 INPUT #9,hsp(t)                   [60F2]
59768 INPUT #9,h3$(t)                   [DBDE]
59770 NEXT t                            [60FE]
59780 CLOSEIN                           [6C74]
59790 RETURN                            [031C]
59795 REM *** Highscores abspeichern ***
                                        [0C4C]
59800 OPENOUT "softchef.hsc"            [6FE4]
59810 FOR t=1 TO 8                      [7856]
59820 WRITE #9,hsf(t)                   [23C2]
59830 WRITE #9,hsff(t)                  [5B90]
59840 WRITE #9,hsw(t)                   [0EE8]
59850 WRITE #9,hswf(t)                  [2DB6]
59860 WRITE #9,hswt(t)                  [29D4]
59862 WRITE #9,h1$(t)                   [D8C6]
59864 WRITE #9,h2$(t)                   [D2CC]
59866 WRITE #9,hsp(t)                   [09EA]
59868 WRITE #9,h3$(t)                   [1CD6]
59870 NEXT t                            [6E00]
59880 CLOSEOUT:a$="softchef.bak":|ERA,@a
      $                                 [8F12]
59890 RETURN                            [BC1E]
60000 DATA Eddivision,Antirock,Bruderbun
      d,Datahard,Electronic Cars,Elise,E
      ssix,Sydney House,Sublogik        [E95C]
60010 DATA A view to a bill,14.25,Alcatr
      az,16.5,Arschon,20.25,Arschon II,1
      4,Ballblaeser,17.25               [2C6C]
60020 DATA Volleyball,15.25,Strand Kopf,
      13,Strand Kopf II,14,Blauer Max,13
      .25,Shoulder Crash,18.75          [FE2C]
60030 DATA Shoulder Crash II,18.75,Bruce
       Klee,17.25,Bac Rogers,10.5,Burger
      queen,10.75,Burning Schrubber,12.7
      5                                 [6BD0]
60040 DATA Chip - Laster,7.75,Willi's Tu
      erme,8.5,Krystle's Castles,17,Elfk
      ampf,17.25,Dick & Doof,13.75      [707C]
60050 DATA Diners Eggs,15.75,Exploding M
      ist,15.75,Exploding Mist II,16.75,
      Leiter Pilot,16.25,Flapp and Flopp
      ,18.75                            [50C0]
60060 DATA Foodball Manager,20,Trostbust
      ers,17,Grundmaster,13,Gratrenner,9
      .25,Gyros,9.25                    [0E0C]
60070 DATA Nero,13.75,Hessengames,15.5,H
      oechste Zeit,6.5,Hunchcrack,7.5,Ba
      stler 64,7.75                     [84EA]
60080 DATA Imbossible Mission,23.5,Int.
      Dennis,13,Jumpwoman,15,Keiser,19.2
      5,Keramika,19.75                  [DF48]
60090 DATA Frankfurt Approach,17.5,Kikst
      op,19,Bloeder Runner,16.5,Master o
      f the Champs,21.25,Matsch Point,20
      .5                                [6C4C]
60100 DATA Rotor Mania,10.25,Mrs. Robot,
      19.5,Coal's Well,19,In-Fort-Tennis
      ,21,Three-on-Three,19.5           [2EDA]
60110 DATA Pfeifenlinie,20,Pitstart,18.7
      5,Pitstart II,23,Coal Position,19.
      5,Poster Zaster,16.75             [516C]
60120 DATA W-Bert,10.75,Rest for Tires,1
      7.25,Heptan,9,Mutant Lords,8,Shive
      r Raid,7.25                       [9CBE]
60130 DATA Globotron,7.25,Strampel,5,Ter
      pentin,12.5,Primus,12,Ski-Weltclub
      ,7.75                             [F650]
60140 DATA Hellfox,18.25,Schlappschwanz,
      12.5,Schlumpf II,7,Smokie,19.25,Po
      ker II,16.5                       [FC9E]
60150 DATA Poker III,16,Spass Pilot,14.7
      5,Spass Taxi,21.75,Spei & Spei,18.
      5,Star Dreck,11                   [0418]
60160 DATA Bella 7,20.5,Strip Skat,18.75
      ,Sublogik Light II,21.5,Sommer Gam
      es,21.75,Sommer Games II,23.75    [CC4A]
60170 DATA Superstrampel,8.25,Super Hexe
      n,14,Ferrori,6.5,The Denver Quest,
      18.25,Tragik,19.75                [66C2]
60180 DATA Tran,6,Turbo Diesel 64,16,Ab'
      n Daun,18.75,Garlok,10.75,Kissensc
      hlacht,6                          [969C]
60190 DATA Winter Shames,23.75,Blizzard,
      14.5,Blizzard of Wor,10,Saga,15,Br
      uchbude,12.75                     [4E98]
60200 DATA Hexen,13.75,Seppel,15.5,Kodia
      c,10.5,Bone Six,6.5,Facing destr.
      Set,22.5                          [83A2]
```

**Listing 2. Erobern Sie den Software-Markt (Schluß)**

# Locomotives Basic-Spezialitäten

**Eine Besonderheit des Betriebssystems Ihres Schneider CPC ist die Verwaltung von Interrupts. Doch die meisten Besitzer wissen nicht so recht, was man mit der Interruptsteuerung überhaupt anfangen kann. Hier erfahren Sie alles über die Programmierung von Interrupts im Locomotive-Basic.**

Was sind eigentlich Interrupts? Kaum ein Schlagwort in der Programmierszene ruft soviel Faszination bei gleichzeitiger Unwissenheit hervor. Dabei ist eigentlich alles ganz einfach.

Die inneren Arbeitsvorgänge in einem Computer sehen normalerweise so aus: Das Gerät hat ein festes Programm, das die CPU (beim Schneider CPC der Z80) unablässig abarbeitet. Der Mikroprozessor holt sich ständig einen Befehl nach dem anderen aus dem Speicher und bearbeitet ihn. Kein äußeres Ereignis kann den Computer erschüttern oder ihn gar von seiner Arbeit abhalten. Dem Benutzer ist es höchstens vorbehalten, die Maschine abzuschalten, um diese kontinuierliche Arbeit zu stoppen. Was passiert nun aber, wenn der Benutzer beispielsweise einen Reset auslöst, etwa beim Schneider CPC mit den Tasten <CTRL + SHIFT + ESC>? Wie Sie sicher wissen, unterbricht der Computer ja in diesem Fall die Bearbeitung des gerade laufenden Programms und setzt alle Software-Einheiten und Hardware-Bausteine zurück. Er verhält sich so, als ob die Stromversorgung gerade eingeschaltet wurde. Scheinbar gibt es also doch einen Weg, den Computer von seiner Arbeit abzuhalten. Und damit sind wir bei dem Thema »Interrupt« – zu Deutsch »Unterbrechung« angelangt.

Ein Reset-Schalter (den Ihr Schneider übrigens normalerweise nicht besitzt) löst fast immer einen Hardware-Interrupt aus. In diesem Fall weiß der Computer von der potentiellen Möglichkeit eines Interrupts so lange nichts, bis dieser tatsächlich auftritt. Dann werden zwei Leitungen des Mikroprozessors zusammengeschaltet, worauf die Hardware-Logik den Computer zurücksetzt. Beim Z80 muß dazu Pin 41 (RESET) auf Masse (Pin 2 oder 49) gelegt werden. Dies ist die Urform des Interrupts.

Eine Abwandlung dieser Form, die auch der Schneider benutzt, ist das sogenannte »Polling«. Hier weiß das Betriebssystem schon vorher, daß ein Interrupt auftreten darf. Deshalb muß dieser nicht mehr den Prozessor vollständig anhalten, sondern braucht sich nur auf irgendeine Weise bemerkbar machen. Dazu muß er nun in einem Port des Z80 oder einer Speicherstelle eine »Fahne«, ein sogenanntes »Flag«, aufstellen. Damit der Computer sofort ohne Zeitverzögerung merkt, ob eine Interruptanforderung besteht, schaut er regelmäßig nach, ob ein solches Flag aktiv ist. Dieses regelmäßige Nachsehen heißt »Polling« und wird von dem Betriebssystem mehrmals jede Sekunde durchgeführt.

## Bis hierher und nicht weiter

Beim Schneider CPC und einigen anderen Computern verhält es sich so, daß eine bestimmte Routine, oft »ISR« oder »Interrupt Service Routine« genannt, ständig bearbeitet wird. Sie hat nicht nur die Aufgabe, Interruptanforderungen zu prüfen, sondern muß auch regelmäßig andere kleine Maschinenroutinen berücksichtigen.

Dazu gehört unter anderem die Tastaturabfrage. Um das zu testen, geben Sie folgendes Programm ein:

```
10 FOR i=1 TO 10000:NEXT i
```

Drücken Sie nun während dem Lauf dieser Leerschleife einige Tasten, so sehen Sie, daß der Computer diese sich intern merkt und auf dem Bildschirm ausgibt, sobald das Programm beendet ist und »Ready« zu lesen ist. Bis zu 20 Zeichen kann Ihr Schneider intern so zwischenspeichern. Doch warum ist das so? Die Tastatur wird natürlich auch während der Arbeit mit dem Programm durch Interrupt abgefragt und die Ergebnisse dieser Abfrage zwischengespeichert. Dieses Verfahren nennt man »interruptgesteuerte Tastaturabfrage«.

Auch fünf verschiedene Zeitgeber werden mittels Interruptroutinen ständig auf aktuellem Stand gehalten. Einen dieser Timer fragt man mit der Basic-Funktion »TIME« ab. Die anderen vier dienen der Programmierung von Interrupts unter Basic.

Interrupts sind also Eingriffe in die Arbeit des Computers, rein auf Grund der vergangenen Zeit unabhängig von dem Programm. Was ist dann aber der Unterschied zum »Multitasking«, das ja gerade mit dem DOS Plus des Schneider PCs immer mehr zum Thema wird?

Interrupts, wie sie der Schneider CPC und die meisten anderen Heimcomputer benutzen, unterbrechen Hauptprogramme, beispielsweise einen Basic-Interpreter, ein Textprogramm oder auch nur den Kommandoprozessor des Betriebssystems (dieser bringt beispielsweise den A>-Prompt von CP/M auf den Bildschirm). Nebenher laufen ständig kleine Programme ab, die sogenannten Interruptroutinen. Grafisch läßt sich die Verteilung der Rechenzeit so darstellen:

```
Hauptprogramm: -  ---  --  --  --  -  -
Interrupt 1:        -   -   -   -   -
Interrupt 2:      -    -    -    -
Interrupt 3:      -         -        -
```

Sie sehen, daß das Hauptprogramm immer langsamer wird, je mehr Unterprogramme durch Interrupt aufgerufen werden.

Beim Multitasking hingegen laufen zwei oder mehr Programme quasi parallel ab. Es werden also mehrere komplette Programme scheinbar gleichzeitig bearbeitet. So kann zum Beispiel auf dem Amiga von Commodore ein Textverarbeitungsprogramm, ein Spiel, und ein Demonstrationsprogramm nebeneinander ablaufen:

```
Textverarbeitung: ---------------------
Spiel:            ---------------------
Demo:             ---------------------
```

Natürlich verarbeitet der Computer auch im Multitasking-Betrieb nicht mehrere Programme tatsächlich gleichzeitig. Der Trick besteht hier darin, daß ein Verteilerprogramm, der sogenannte »Dispatcher«, eine gewisse Rechenzeit auf den einzelnen Programmen zuordnet. Er teilt jedem der einzelnen Routinen für Sekundenbruchteile die komplette Prozessorleistung zu. Im allgemeinen erhalten die Programme der Reihe nach jeweils einen solchen »Zeitschlitz«. Falls eine Priorität gesetzt wird (wie bei allen Großrechenanlagen), ist dieser Zeitschlitz entsprechend größer oder ein bestimmtes Programm wird häufiger aufgerufen. Besitzt das Hauptprogramm die vierfache Priorität und sind zwei Unterprogramme gleich wichtig, dann kann die Prozessorleistung beispielsweise so aufgeteilt werden.

```
Hauptprogramm: -- -- -- -- -- -- -- --
Unterprogramm:  -   -   -   -   -
Unterprogramm:    -   -   -   -
```

Es ist aber auch

```
Hauptprogramm: --- --- --- --- ---
Unterprogramm:   -  -  -  -  -
Unterprogramm:    -  -  -  -
```

möglich.Die hohe Umschaltgeschwindigkeit hinterläßt beim Betrachter den Eindruck, als ob mehrere Programme gleichzeitig verarbeitet würden.

## EVERY und AFTER

Die beiden wichtigsten Befehle für das Programmieren von Unterprogrammen im Interruptverfahren, heißen »EVERY« und »AFTER«.

Die Syntax der beiden Befehle ist identisch:

```
EVERY Zeitwert,Timer GOSUB Zeile
AFTER Zeitwert,Timer GOSUB Zeile
```

Der Unterschied zwischen den beiden Befehlen liegt auf der Hand: Während bei AFTER das Unterprogramm nach Verstreichen des Zeitintervalls ein einziges Mal aufgerufen wird, führt der Basic-Interpreter bei EVERY die gewünschte Befehlsfolge regelmäßig aus, bis die Interruptsteuerung wieder abgeschaltet wird.

Das Locomotive-Basic ruft Interruptroutinen alle 0,02 Sekunden, also jede 1/50 Sekunde, auf. Der »Zeitwert« ist daher ein Vielfaches von 0,02 Sekunden. Wollen Sie ein Unterprogramm genau einmal pro Sekunde aufrufen, funktioniert das unter Basic folgendermaßen:

```
10 EVERY 50 GOSUB 100
```

Der Timer muß nicht unbedingt angegeben werden. Falls Sie ihn weglassen, wird automatisch der Wert 0 benutzt.

Damit aus der Anweisung ein vollständiges Programm wird, dürfen Sie die eigentliche Interruptroutine natürlich nicht vergessen. Sie wird, wie ein ganz normales Unterprogramm, das man mit »GOSUB« aufruft, programmiert.

```
10 EVERY 50 GOSUB 100
20 GOTO 20
100 PRINT "Der Interrupt ..."
110 RETURN
```

Die Programmzeile 20 ist notwendig, da sonst der Basic-Interpreter irrtümlich in die Interruptroutine »hineinläuft«. Das führt dann zu der Fehlermeldung

```
Unexpected RETURN
```

Jetzt fragen Sie wahrscheinlich, was nun eigentlich die Bedeutung dieser Interruptsteuerung ausmacht. Der Computer scheint ja doch nicht zwei Basic-Programme gleichzeitig abzuarbeiten, sondern die Ausgabe des Textes »Der Interrupt ...« wird nur in ein bestimmtes Zeitraster gezwängt.

Falsch – und ob der Computer zwei Programmteile gleichzeitig bearbeitet! Es gibt nämlich noch ein Hauptprogramm. Es besteht zwar nur aus der Zeile

```
20 GOTO 20
```

aber es ist vorhanden. Dem läßt sich abhelfen. Schalten Sie einfach mit

```
TRON
```

das Auflisten der bearbeiteten Zeilennummern ein. Dann sehen Sie die tatsächliche Programmstruktur.

Das folgende Programm gibt alle Aktivitäten – sowohl des Unterprogramms als auch des Hauptprogramms – auf dem Bildschirm an:

```
10 EVERY 50 GOSUB 40
20 PRINT "Das Hauptprogramm ..."
30 GOTO 20
40 PRINT "Der Interrupt ..."
50 RETURN
60 END
```

Die Anzeige

```
Das Hauptprogramm
```

wird regelmäßig von der Meldung

```
Der Interrupt
```

unterbrochen.

Dabei wird das Hauptprogramm um so langsamer, je umfangreicher die Interruptroutinen sind. Deshalb gilt die Regel, diese Routinen immer so kompakt wie möglich zu schreiben. Nur dann wird das Hauptprogramm durch sie nicht allzu lange aufgehalten.

Daneben ist es wichtig, den Zeitwert geschickt zu wählen. Ändern Sie im obigen Programm die Zeile 10 in

```
10 EVERY 1 GOSUB 40
```

und das Hauptprogramm kommt überhaupt nicht mehr zum Zuge. Die Interruptroutine wird so oft aufgerufen, daß für andere Arbeiten keine Zeit mehr bleibt.

Experimentieren Sie deshalb mit dem Zeitwert so lange herum, bis Sie eine vernünftige Zeitaufteilung zwischen Interrupt- und Hauptprogramm finden.

## Interrupts sind gefährlich

Auch wenn Sie es vielleicht noch nicht so recht glauben wollen, Interrupts sind eine gefährliche Sache. Nicht etwa, weil sie selbst fehlerhaft sind oder das Locomotive-Basic die Interrupts fehlerhaft bearbeitet! Nein, diesmal ist der Mensch, sprich der Programmierer, selbst schuld. Es liegt in der Natur des Menschen, komplexe Vorgänge, besonders wenn sie wechselnde Startbedingungen besitzen, nur schwer nachvollziehen zu können. Sie glauben gar nicht, was für haarsträubende Situationen Interrupts teilweise verursachen...

Ein Beispiel:

```
10 EVERY 5 GOSUB 50
20 PRINT "Das ";
30 PRINT "Hauptprogramm ...";
40 GOTO 20
50 PRINT "Der Interrupt ...";
60 RETURN
70 END
```

An sich sollte die Bildschirmausgabe dieses Programms mit der des vorherigen Beispiels identisch sein. Aber spätestens, wenn Sie das Programm starten, erleben Sie eine (unangenehme) Überraschung. Auf dem Bildschirm erscheint:

```
Das Hauptprogramm ...
Das Der Interrupt ...
Hauptprogramm ...
Das Der Interrupt ...
Hauptprogramm ...
```

Genau zwischen die Zeilen 20 und 30 »platzt« der Interrupt hinein und zerstört die formatierte Bildschirmausgabe.

Eine ganz besondere Tücke entdecken Sie, wenn Sie einmal <ESC> drücken und damit das laufende Programm anhalten. Sobald Sie eine andere Taste betätigen und den Computer weiterarbeiten lassen, versucht das Betriebssystem, die verlorengegangenen Interruptintervalle »nachzuholen«. Das trifft auch für den Fall zu, daß Sie die Bearbeitung des Programms abbrechen und später mit »CONT« fortsetzen oder mit »INKEY$« den Computer auf eine Eingabe warten lassen:

```
Der Interrupt ...
Der Interrupt ...
Der Interrupt ...
```

ist das Ergebnis. Das Hauptprogramm wird auf längere Zeit vollständig eingefroren, bis der Computer mal wieder Zeit hat, sich diesem zu widmen.

Daß in dem Fall natürlich die Programmsteuerung völlig durcheinandergerät, ist klar. Positive Auswirkungen zeigt dieser »Nachholversuch« nur bei Uhrenprogrammen, die damit trotz Programmunterbrechung die korrekte Zeit anzeigen. Haben Sie allerdings ein Programm geschrieben, bei dem Hauptprogramm und Interrupt direkt zusammenarbeiten und gar über festgelegte Kanäle Daten austauschen, führt diese Eigenheit ins Chaos. Denn die Interruptroutine stellt dann berechnete Werte zur Verfügung, die das Hauptprogramm im Moment noch gar nicht abholen will.

Was ist nun die »Moral von der Geschichte«? Interruptroutinen laufen erstens anders und zweitens als man denkt. Machen Sie sich auf die unmöglichsten Situationen gefaßt! Sind Sie aber bereit, sich mit diesen »Macken« abzufinden, erzielen Sie mit interruptgesteuerten Programmen tolle Effekte.

Wie steht's zum Beispiel mit dieser kleinen, aber nichtsdestoweniger sehr



nützlichen Routine? Computer wie der IBM-PC und Atari-ST besitzen eine Taste, mit der der Bildschirminhalt auf dem Drucker ausgegeben wird. So etwas macht sich natürlich auch beim Schneider CPC gut. Der Einfachheit halber schreibt unsere Routine den Bildschirminhalt jedoch lediglich auf die Diskette beziehungsweise Kassette. Setzen Sie in Zeile 120 statt des Befehls »SAVE« eine GOSUB-Anweisung auf eine Hardcopy-Routine (zum Beispiel der auf Seite 74 aus Happy-Computer, Ausgabe 12/85 oder auf Seite 80 aus Happy-Computer, Ausgabe 6/86).

Als Taste zum Programmaufruf dient <COPY>. Sie besitzt den Code 9 (siehe Handbuch), der mit INKEY abgefragt wird. Die Interruptroutine besteht im wesentlichen aus der Abfrage dieser Taste. Wurde sie nicht gedrückt (wird also der Wert -1 zurückgegeben), so soll der Basic-Interpreter ohne weitere Aktion aus dem Unterprogramm zurückkehren. Ansonsten soll der Inhalt des Bildschirms gesichert werden. Das Hauptprogramm unseres Beispiels zeichnet Linien, deren Aussehen der Zufallsgenerator steuert:

```
10 EVERY 50 GOSUB 100
20 MOVE RND*640,RND*400
30 DRAW RND*640,RND*400
40 GOTO 20
100 if INKEY(9)=-1 THEN RETURN
110 SAVE "!SCREEN.SRN",b,&C000,
&4000
120 RETURN
130 END
```

Dagegen gibt es allerdings Programmteile, in denen keine Interrupts auftreten dürfen. Dazu gehören zum Beispiel Routinen, die Grafik auf dem Bildschirm ausgeben. Benutzt nun auch die Interruptroutine Grafikbefehle, kommt es zu Fehlern. Das folgende Programm arbeitet noch einwandfrei:

```
10 CLS
30 FOR i=1 TO 640 STEP 4
40 MOVE i,0
50 DRAWR 0,20
60 NEXT i
70 END
```

Bauen Sie nun aber eine Interruptroutine ein, die mit »MOVE« den Grafikcursor an eine zufällige Position auf dem Bildschirm setzt, so ist das Ergebnis verheerend:

```
10 CLS
20 EVERY 5 GOSUB 80
30 FOR i=1 TO 640 STEP 4
40 MOVE i,0
50 DRAWR 0,20
60 NEXT i
70 END
80 MOVE RND*640,RND*400
90 RETURN
```

Falls der Interrupt direkt nach Zeile 40 auftritt, werden eine – oder auch mehrere – Linien falsch positioniert. Die in Zeile 40 festgelegte Position des Grafikcursors wird nämlich in Zeile 80 überschrieben. Der Computer arbeitet in Zeile 50 weiter und zeichnet die Linie an die falsche Stelle.

Man muß nun auf irgendeine Art und Weise verhindern, daß der Computer während dem Bearbeiten der Zeilen 30 und 40 die Interruptanforderung beachtet. Keine Angst, auch dies ist im Schneider-Basic vorgesehen. Der Befehl zum Sperren von Interrupts heißt »DI«. DI steht für die englische Bezeichnung »Disable Interrupts«. Der Befehl »EI« (»Enable Interrupts«) hebt die Sperre wieder auf.

Erweitern wir also unser kleines Programm um diese beiden Befehle:

```
10 CLS
20 EVERY 5 GOSUB 80
30 FOR i=1 TO 640 STEP 4
35 DI
40 MOVE i,0
50 DRAWR 0,20
55 EI
60 NEXT i
70 END
80 MOVE RND*640,RND*400
90 RETURN
```

Nach dieser kleinen Änderung läuft das Programm wieder ohne Störungen.

Interruptroutinen sind übrigens gegen Selbstaufruf »immun«. Das heißt, während ein Unterprogramm interruptgesteuert bearbeitet wird, kann der Computer es nicht erneut aufrufen. Rekursives Basic gibt es also auch beim Schneider CPC nicht.

Manche Systemroutinen des Schneider CPC lassen aus zeitlichen Gründen keine Interrupts zu. Dazu gehört beispielsweise der komplette Datentransfer mit dem Kassettenrecorder oder der Diskettenstation. Interrupts würden das Taktraster, mit dem Daten gelesen oder geschrieben werden, derart stören, daß Übertragungsfehler auftreten würden. Folgendes kleine Programm zeigt die Behandlung von Interrupts während des Datentransfers:

```
10 | TAPE
20 EVERY 10 GOSUB 100
30 FOR i=1 TO 2000:NEXT i
40 CAT
50 END
100 PRINT CHR$(7);
110 RETURN
```

Mit einem Vortex-Controller müssen Sie Zeile 10 durch

```
10 | CAS
```

ersetzen. Ohne Diskettenstation streichen Sie Zeile 10 ersatzlos.

Während der FOR-NEXT-Schleife in Zeile 30 sind Interrupts zugelassen. Der Lautsprecher gibt in unserem Beispiel Pfeiftöne ab, die ein Interrupt in Zeile 100 auslöst. Sobald aber der CAT-Befehl in der Programmzeile 40 bearbeitet wird, schaltet sich der Motor des Kassettenrecorders ein. Das Betriebssystem läßt ab diesem Zeitpunkt keine Interrupts mehr zu – die Tonausgabe unterbleibt.

Auch während der Bearbeitung verschiedener Diskettenroutinen, wie dem Lesen, Schreiben und Formatieren von Sektoren, werden keine Interrupts bearbeitet. Da diese Basisroutinen auch von höherstehenden ROM-Routinen aufgerufen werden, sind zum Beispiel auch während »|DIR« und »CAT« keine Interrupts erlaubt.

## Ein Interrupt, zwei Interrupts...

Falls Sie langsam an diesem Konzept Spaß finden, kommen Sie mit einer einzigen Interruptroutine nicht ganz aus. Das berücksichtigt auch Locomotive. Das Schneider-Basic gesteht Ihnen deshalb das Bearbeiten von bis zu vier Interruptroutinen zu.

Damit aber der Interpreter weiß, welchen der vier Interrupt-Zeitgeber Sie ansprechen wollen, muß in der Syntax der Wert »Timer« angegeben werden. »Timer« darf die Werte 0 bis 3 annehmen – entsprechend den gewünschten Interrupt-Zeitgebern. Falls Sie keinen Wert angeben, wie wir das bisher immer gemacht haben, setzt der Basic-Interpreter (wie oben erwähnt) automatisch den Timer 0.

Folgendes Basic-Programm initialisiert alle vier Interrupt-Zeitgeber. Das Hauptprogramm bringt lediglich Punkte auf den Bildschirm, die Interruptroutinen 0 bis 3 drucken ihre Kennummer aus:

```
10 EVERY 10 GOSUB 100
20 EVERY 20,1 GOSUB 200
30 EVERY 40,2 GOSUB 300
40 EVERY 80,3 GOSUB 400
50 PRINT ".";
60 GOTO 50
100 PRINT "0";
110 RETURN
200 PRINT "1";
210 RETURN
300 PRINT "2";
310 RETURN
400 PRINT "3";
410 RETURN
420 END
```

Alle zehn Takte wird eine »Null« auf dem Bildschirm ausgegeben, halb so oft eine »1«. Alle 40 Takte erscheint die »2« und alle 80 Takte die »3«.

Ganz klar verdeutlicht die Bildschirmausgabe dieses Programms die unterschiedliche Priorität der Interruptroutinen. Darunter versteht man die Entscheidung des Computers, welche Routine bevorzugt wird, wenn zwei

Interrupt-Anforderungen zum gleichen Zeitpunkt anfallen. Der Interrupt mit der höchsten Nummer, also 3, besitzt auch die höchste Priorität. Die zweithöchste Priorität besitzt der Timer 2, dann kommt der Timer 1 und zu guter Letzt der Zeitgeber 0. Somit zeigt das Programm nach jeweils acht Aufrufen der Interruptroutinen die Zeichenkombination »3210« an – und nicht etwa »0123« oder gar »1032«.

Ein Interrupt wird vom Basic-Interpreter selbsttätig abgemeldet, wenn das Hauptprogramm beendet ist. Immer wieder ist es allerdings notwendig, Interruptroutinen auf einzelne Teile des Hauptprogramms zu beschränken. Zum Abschalten eines Interrupts dient »REMAIN«:

```
PRINT REMAIN(Timer)
```

oder

```
x=REMAIN(Timer)
```

Die Funktion REMAIN hat eigentlich zwei Aufgaben. Zuerst einmal deinstalliert sie die angegebene Interruptroutine. Als Nebeneffekt gibt sie aber eine Zahl zurück, die die Anzahl der Timerimpulse angibt, die bis zum nächsten Auftreten des Interrupts noch nötig sind. Als Parameter erwartet REMAIN die Nummer des betroffenen Zeitgebers:

```
dummy=REMAIN(0)
x=REMAIN(2)
PRINT REMAIN(1)
```

## Der AFTER-Befehl

Kommen wir zu dem zweiten Befehl, der Interruptroutinen aktiviert. Eigentlich ist ein mit »AFTER« aufgerufenes Unterprogramm gar kein richtiges Interruptprogramm. Denn es wird nicht regelmäßig abgearbeitet, sondern nur ein einziges Mal – am Ende der vorher angegebenen Zeit.

Sinnvolle Anwendungen für die Anweisung »AFTER« zu finden, ist dann auch etwas schwieriger als bei »EVERY«. Wie steht es aber mit einem Wecker, der nach zehn Minuten einen Warnton ausgibt?

```
10 AFTER 30000 GOSUB 30
20 GOTO 20
30 PRINT CHR$(7);"Zeitlimit
erreicht!"
40 END
```

Auch der AFTER-Befehl greift auf die insgesamt vier Zeitgeber des Betriebssystems zu:

```
AFTER 1000,0 GOSUB 1000
AFTER 2000,1 GOSUB 2000
AFTER 3000,2 GOSUB 3000
AFTER 4000,3 GOSUB 4000
```

Unabhängig vom Befehl dürfen Sie hier jede Timernummer nur ein einziges Mal vergeben. Haben Sie also bereits einen »EVERY«-Interrupt mit dem Timer 2 initialisiert, dürfen Sie diesen Zeitgeber nicht mit »AFTER« noch einmal benutzen.

Die maximale Zeitspanne, die bei »AFTER« und bei »EVERY« angegeben werden darf, beträgt 32 767 Zeiteinheiten. Das sind immerhin zirka 655 Sekunden oder knapp elf Minuten.

## Kräftiger Sound mit ON-SQ-GOSUB

So praktisch die Programmierung von Interruptroutinen für allgemeine Zwekke ist, so unbrauchbar erweist sich das Prinzip bei der Wiedergabe von musikalischen Effekten. Denn Interrupts werden in einem festen Taktraster aufgerufen, während Musikstücke Noten von verschiedener Länge enthalten. Das bringt entweder die Interrupts oder die klangliche Stimmigkeit der Musik durcheinander.

Aber auch das haben die Programmierer von Locomotive erkannt und deshalb das Konzept der Warteschlangen entwickelt.

Damit kann der Computer mehrere Töne intern zwischenspeichern, bevor er sie an den Soundchip ausgibt. So interpretiert der Computer schon neue Basic-Befehle, während der Tongenerator noch Töne ausgibt.

Leider hat dieses Konzept eine entscheidende Schwachstelle: Eine Warteschlange kann pro Tonkanal maximal vier Töne (oder Geräusche) aufnehmen. Deshalb mußte Locomotive noch einmal in die Software-Trickkiste greifen und den Sprachumfang des Basic-Interpreters um den Befehl

```
ON SQ(Tongenerator) GOSUB Zeile
```

erweitern. Nun ist es tatsächlich möglich, gleichzeitig Musikstücke zu spielen und ein Basic-Programm zu bearbeiten. Mit Hilfe von »EVERY« kann man das Betriebssystem sogar veranlassen, ein Hauptprogramm, vier Interruptroutinen und das Musikstück vom Computer quasi parallel zu bearbeiten.

Sehen wir uns einmal ein Programm an, das den SOUND-Befehl benutzt:

```
10 FOR i=1 TO 200
20 SOUND 1,i
30 NEXT i
```

Diese drei Zeilen arbeiten noch ohne Interrupts. Wir stellen allerdings dem Computer die Aufgabe, im Hauptprogramm Linien zu zeichnen, ohne die Tonausgabe zu vergessen.

Dazu brauchen wir zuerst einmal die exakte Definition der Funktionsweise von »ON SQ(x) GOSUB«. Der Befehl weist den Interpreter an, das angegebene Unterprogramm aufzurufen, sobald die Tonwarteschlange – für den als Argument in Klammern übergebenen Soundkanal – leer ist. Dabei verhält sich »ON SQ« ähnlich wie »AFTER« – gegensätzlich zu »EVERY« also. Sobald das Unterprogramm einmal abgearbeitet ist, wird nämlich der Interrupt gelöscht.

```
100 MODE 2
110 ON SQ(1) GOSUB 170
120 ' Hauptprogramm *************
130 MOVE RND*640,RND*400
140 DRAW RND*640,RND*400
150 GOTO 130
160 ' Soundroutine **************
170 i=i+1
180 SOUND 1,i
190 ON SQ(1) GOSUB 170
200 RETURN
210 END ' ***********************
```

Das Hauptprogramm besteht lediglich aus »DRAW« und »MOVE« mit zufallsbedingten Werten, sowie dem GOTO-Befehl in der Zeile 150, der für die permanente Wiederholung des Programms sorgt.

Da bereits nach Zeile 110 die Warteschlange des Tongenerators leer ist, ruft das Programm die Soundroutine in Zeile 170 auf. Dort wird der Computer endlich angewiesen, einen Ton auszugeben. In Zeile 190 initialisiert das Programm (wie oben gefordert) den Interrupt neu. Dem Benutzer scheint es, als ob die Tonausgabe gleichzeitig mit dem Zeichnen der Linien erfolgt. Wenn Sie nun wissen wollen, wie der Basic-Interpreter die Rechenzeit auf das Hauptprogramm und die Tonausgabe aufteilt, tippen Sie das folgende Programm ab. Es zeigt im Hauptprogramm ständig die Meldung »Happy« an. Sobald die Tonroutine aufgerufen wird, bringt diese bei jedem Durchlauf den invers dargestellten Text »Computer« auf dem Bildschirm:

```
100 MODE 2
110 i=20
120 ON SQ(1) GOSUB 150
125 ' Hauptprogramm *************
130 PRINT " Happy  ";
140 GOTO 130
145 ' Tonausgabe ****************
150 i=i+1
160 SOUND 1,i
170 PRINT CHR$(24);" Computer ";
    CHR$(24);" ";
180 ON SQ(1) GOSUB 150
190 RETURN
```

Der Schneider CPC besitzt schon unter Basic hervorragende Befehle für Interrupts. Damit zu programmieren gestaltet sich bedeutend einfacher als mit schwerfälligen Ersatzmethoden, die bei anderen Computern für solche Zwecke erforderlich sind. Unter Umständen ist der Einsatz aber auch sehr gefährlich. Also – machen Sie sich die Wirkungsweise der Interrupts vorher genau bewußt, sonst gibt es Pannen.

(Martin Kotulla/hg)

# Interrupts – Programmieren mit Pfiff

**Der Schneider CPC verfügt über eine ausgefeilte Interruptstruktur. Diese läßt sich nicht nur von Basic-Programmen aus nutzen, sondern auch in Maschinensprache.**

Interrupt – das heißt Programmieren abhängig nur von der Zeit und nicht vom Programmaufbau. Wie das unter Basic geht, darauf gingen wir ausführlich im Beitrag auf Seite 116 ein. Doch auch in Maschinencode ist es nicht schwer.

Der konzeptionelle Aufbau der Interruptsteuerung ist in Maschinensprache der gleiche wie der der Basic-Befehle EVERY, AFTER und ON SQ. Auch in Maschinensprache lassen sich Interruptroutinen in die Zeitschleife »einklinken« und aus der Warteschlange der Interrupts wieder löschen. Dabei reicht die Unterstützung von Maschinenroutinen durch den Teil des Betriebssystems, der Interrupts bearbeitet, sogar noch weiter als unter Basic. So lassen sich verschiedene Typen von Interrupts auswählen. Auch ist es sehr angenehm, daß Interrupts für Maschinenroutinen im Direktmodus des Interpreters weiter beachtet werden. Interruptprogramme in Basic werden ja mit Ende des Hauptprogramms gestoppt.

Die Interruptsteuerung arbeitet Betriebssystem-intern mit einem einzigen Taktraster. Die Hauptschleife des Interrupts wird 300mal in der Sekunde aufgerufen. Von diesem Grundwert leitet der Computer vier verschiedene Interruptarten mit drei Taktrastern ab.

Der schnelle Interrupt (Fast Ticker Chain) wird 300mal pro Sekunde aufgerufen. Er eignet sich für besonders häufige Aufrufe.

Der Sound-Interrupt (Sound Chain) wird 100mal pro Sekunde aufgerufen und dient der Sounderzeugung. Der Sound-Interrupt ist deshalb auch nur dem Betriebssystem zugänglich, nicht aber normalen Programmen.

50mal pro Sekunde wird der Frame-Flyback-Interrupt (Frame Flyback Chain) aufgerufen und damit während jedes Bildrücklaufs des Elektronenstrahls im Monitor.

Der wichtigste Interrupt ist der normale Interrupt (Ticker Chain). Zu ihm zählt beispielsweise die Abfrage der Tastatur. 50mal pro Sekunde wird er aktiviert.

Beim Einsatz von Interrupts in eigenen Programmen verzichten Sie besser auf den erstgenannten Interrupt. Hier in das Betriebssystem eingehängt, werden die Routinen sehr oft aufgerufen und bremsen dadurch die Arbeitsgeschwindigkeit des Computers. Ein normaler Ticker belastet das System im Vergleich zu einem Fast-Ticker mit nur einem Sechstel der Rechenzeit.

Der Bildrücklauf-Interrupt steht zwar zur allgemeinen Nutzung bereit, ergibt aber nur Sinn bei Programmen, die direkt mit dem Bildschirm arbeiten sollen. Dazu gehören etwa die Routinen des Systems, die bewirken, daß Farben blinken oder der Bildschirm flimmerfrei scrollt.

Neben diesen vier Kategorien gibt es noch eine weitere grundsätzliche Einteilung von Interrupts: Sie sind synchron oder asynchron.

## Gefahr durch asynchrone Ereignisse

Asynchrone Interruptroutinen heißen auch »asynchrone Ereignisse« oder auf Englisch »asynchronous Events«. Sie werden streng periodisch aufgerufen – platzen also auch mitten in ein Programm oder werden während dem Bearbeiten einer Routine des Betriebssystems aktiv. Das birgt natürlich die Gefahr, daß das asynchrone Ereignis Dinge tut, die nicht so sehr erfreulich sind. So sind viele Systemroutinen beispielsweise nicht »reentrant« – dürfen sich also nicht selbst aufrufen. Unterbricht aber der Zeit-Interrupt zum Beispiel gerade die Bildschirmausgabe und übergibt er einem asynchronen Ereignis die Kontrolle über den Computer, darf dieses Ereignis die Bildschirmausgabe keinesfalls aufrufen – denn Firmwareroutine TXT OUTPUT ist eben nicht »reentrant«.

Eine besondere Klasse sind allerdings die sogenannten »speziellen« oder »eiligen« asynchronen Ereignisse. Sie werden noch schneller von der Interruptverwaltung aufgerufen als normale asynchrone Events. Ihre Aktivitäten sind aber auch weitaus eingeschränkter. Sie dürfen weder Interrupts freigeben, noch den alternativen Registersatz oder gar die Indexregister IX und IY benutzen. Weiterhin müssen sie so kurz wie möglich sein. Von der Bedingung, daß die Interrupts gesperrt bleiben müssen, leitet sich ferner die Forderung ab, daß keinerlei Restarts und nur ein bestimmter Teil der Systemroutinen benutzt werden dürfen. Denn diese schalten sehr oft die Interrupts wieder ein. Genaue Angaben zu jeder einzelnen Routine des Betriebssystems finden Sie im Firmware-Handbuch von Schneider. Als »Belohnung« für all diese Einschränkungen werden alle asynchron gesteuerten Routinen sehr regelmäßig aufgerufen.

Falls das nicht unbedingt erforderlich ist, arbeiten Sie besser mit synchronen Interrupts. Nach außen hin fällt der Unterschied oft gar nicht auf. In allen anderen Maschinencode-Programmen dürfen Sie aber weiter alle Ressourcen des Computers ausnutzen.

Der Unterschied zwischen synchronen und asynchronen Ereignissen liegt allein im Aufruf. Synchrone Interrupts werden in eine Warteschlange eingereiht. Das Hauptprogramm fragt regelmäßig an, ob neue Routinen zum Bearbeiten vorliegen. Diese Methode kennen Sie bereits als »Polling«. Sie bietet den Vorteil, daß die Interrupts das Hauptprogramm zu einem passenden Zeitpunkt unterbrechen.

Der Basic-Interpreter von Locomotive beachtet das Polling automatisch. Die zugehörige Maschinenroutine liegt beim CPC 464 im Basic-ROM zwischen den Adressen C807 und C844hex. Beim CPC 664 lautet die Startadresse C8B5hex und beim CPC 6128 C8B2hex. Solange als Hauptprogramm eine vom Basic-Interpreter bearbeitete Routine läuft, brauchen Sie sich darum nicht kümmern. Anders liegt der Fall bei einer selbstprogrammierten Maschinencode-Routine. Jetzt muß das Polling »von Hand« eingebaut werden. Ein Beispiel für solch eine Routine finden Sie im folgenden:

```
KL_POLL_SYNC  EQU  &B921
KL_NEXT_SYNC  EQU  &BCFB
KL_DO_SYNC    EQU  &BCFE
KL_DONE_SYNC  EQU  &BD01

POLL:         CALL KL_POLL_SYNC
              RET  NC

POLL2:        CALL KL_NEXT_SYNC
              RET  NC
              PUSH AF
              PUSH HL
              CALL KL_DO_SYNC
```

# Die Schneider-Sonderhefte von Happy-Computer: eine runde Sache




**Schneider 1**

Alle Schneider-Computer im Vergleich. Grafik: »Geheimcodes« zur Bildschirmgestaltung. Listing: Malen wie auf einer Leinwand. Musik und Sound selbst programmieren. Anwendungen: Echtzeitverarbeitung auf dem Schneider/Assembler-Disassembler für den CPC 464.




**Schneider 2**

Wichtige Tips & Tricks für Anfänger und Fortgeschrittenene. Grundlagen: So programmiert man 3D-Grafik. Die interessantesten Firmware-Routinen. Preiswert selbstgebaut: RS232-Schnittstelle mit maßgeschneidertem DFÜ-Programm. Hardware-Einkaufstips: Drucker, Floppy-Laufwerke und Speichererweiterungen.










**Schneider 5**

Lernen Sie den ersten Personal Computer von Schneider kennen. Wir berichten über die CP/M plus-Funktionen BIOS und BDOS. Weitere Hilfe gibt es mit den Basic-Erklärungen. Ein »Flugzeug in Not« sorgt für ein spannendes Listing. Wir zeigen, wie Sie mit der Programmiersprache »Logo« Musik machen können.

**Schneider 3**

Eine ausführliche Beschreibung der Hardware aller CPC. Der Basic-Kurs für Anfänger hilft bei den ersten Programmierschritten. Fortgeschrittene finden eine Einführung in CP/M. Spiele, Anwendungen sowie Grafik und ein Funktionsplot-Programm gestalten dieses Heft zu einer interessanten und herausfordernden Begleitlektüre.

**Schneider 4**

Einsteigern hilft eine ausführliche Basicprogrammierung sowie Nützliches zu Sound und Grafik auf Schneider CPCs. Ebenso Kaufberatung und Grundlagen zu Diskettenlaufwerken. Wieder gibt es jede Menge Tips & Tricks, Spitzenspiele, Grafik und Anwendungslistings.

**Nutzen Sie die Bestellmöglichkeit der Schneider-Sonderhefte 1 bis 5 mit der eingehefteten Zahlkarte im vorliegenden Sonderheft von »Happy-Computer«!**

```
POP  HL
POP  AF
CALL KL_DONE_SYNC
JP   POLL2
```

Die Routine rufen Sie mit

```
CALL POLL
```

(allerdings nicht unter Basic) auf. Mit

```
CALL KL_POLL_SYNC
```

testet die Routine, ob eine Interrupt-Anforderung vorliegt.

Trifft dies zu, weist KL NEXT SYNC den Computer an, die Startadresse der zu bearbeitenden Routine aus der Warteschlange zu holen. KL DO SYNC ruft dann das Unterprogramm auf. KL DONE SYNC sorgt für das korrekte Ende der Unterbrechung. Die Anweisung JP POLL2 sorgt dafür, daß alle Anforderungen beachtet werden.

## Event- und Ticker-Block

Um eine Interruptroutine richtig zu behandeln, benötigt das Betriebssystem Informationen, beispielsweise über den Programmtyp. Die für das Betriebssystem interessanten Werte liegen in einem sogenannten Event-Block vor.

Die Event-Blocks besitzen ein fest vorgegebenes Format. In den ersten beiden Bytes des Blocks steht ein Verkettungszeiger. Diesen verwaltet das Betriebssystem und er darf auf keinen Fall von Ihrem Programm verändert werden.

Das darauffolgende Byte ist der Zähler, der das Verhältnis zwischen den bereits bearbeiteten und den noch anstehenden Aufrufen der Routine beschreibt. Sobald die Routine angefordert wird, erhöht das Betriebssystem den Zähler – beim tatsächlichen Bearbeiten der Routine wird er um 1 herabgesetzt. Der Wert des Zählers liegt immer im Bereich zwischen 1 und 126.

Spezielle Informationen für die Arbeit der Interruptroutinen zeigen negative Werte dieses Zählers an. Ein Wert zwischen –2 und –128 bedeutet, daß die Unterbrechung inaktiv, aber nicht aus der Interruptkette ausgeklinkt ist. Ein Wert Null ruft das Unterprogramm sofort auf.

Das nächste Byte im Event-Block ist bitweise organisiert. Es beschreibt den Typ der Unterbrechung.

Interruptroutinen dürfen entweder im zentralen 32 KByte großen RAM-Bereich zwischen den Adressen 4000 und BFFFhex oder in einem ROM liegen. Programme im RAM werden ohne Umschalten direkt mit CALL vom Betriebssystem aufgerufen. Interruptroutinen im ROM brauchen Informationen, in welchem ROM das Programm steht. Dazu dient das ROM-Select-Byte. Ist Bit 0 gelöscht, so sucht der Computer im RAM nach der Routine. Ein gesetztes Bit 0 deutet auf den ROM-Bereich. Als Tip für eigene Programme: Belegen Sie soweit wie möglich die RAM-Adressen. Der Aufruf ist so bedeutend schneller. Asynchrone Interruptroutinen müssen dann auch immer in diesem Bereich stehen.

Die Bit 1 bis 4 steuern die Priorität der Aufrufe. Synchrone Anforderungen dürfen mit unterschiedlichen Prioritäten ausgestattet sein. Wenn zwei oder mehrere Ereignisse gleichzeitig angefordert werden, dann hat das Programm mit der höheren Priorität Vorrang.

Bit 5 muß stets auf Null gesetzt sein; Bit 6 ist das Unterscheidungsmerkmal für normale (das heißt 50mal pro Sekunde) und eilige (das heißt 300mal pro Sekunde) Aufrufe. Ist Bit 6 gesetzt, so liegt ein eiliger Aufruf vor. Für normale Ereignisse bleibt das Bit gelöscht.

Bit 7 steuert, ob ein Ereignis als synchron oder asynchron initialisiert wird. Bei gesetztem Bit erfolgt die Unterbrechung asynchron, bei synchronen Ereignissen ist das Bit zurückgesetzt.

Byte 4 und 5 des Event-Blocks enthalten die Startadresse der Interruptroutine. Falls Sie eine ROM-Routine (oft auch »Far Address« genannt) benutzen, muß in Byte 6 die Kennziffer der Speicherbank stehen.

Die restlichen Bytes des Event-Blocks sind von Haus aus unbelegt. Sie können hier eventuelle Daten für das Unterprogramm eintragen.

Bis jetzt haben unsere Interruptroutinen aber noch einen Nachteil. Sie werden nur ein einziges Mal ausgeführt. Danach löscht das Betriebssystem sie wieder aus der Event-Kette. Um die Routinen aber nicht völlig zu vergessen, verwaltet der Computer eine weitere Liste, die »Ticker-Chain« beziehungsweise »Fast-Ticker-Chain«. Auch in ihr werden die Ereignisblöcke (Event-Blöcke) eingetragen.

Die Datenstruktur umfaßt allerdings noch einige weitere Bytes. Diese sind dem Event-Block vorangestellt.

Der Fast-Ticker-Block ist nichts ande-

```
; *********************************************
; *                                           *
; *     FLIMMER.ASM - Demo fuer Interrupts    *
; *                                           *
; *********************************************

                ORG     &A000
                JP      INITTICK

KL_NEW_FAST_TIC EQU     #BCE0           ; KL NEW FAST TICKER
KL_DEL_FAST_TIC EQU     #BCE6           ; KL DEL FAST TICKER
HOLD_VALUE      DEFB    0

; ************* TICKERLISTE ********************************************

TICLST          DEFW    0,0
                DEFB    0,0
                DEFW    0
                DEFS    2

; ************* INTERRUPT INITIALISIEREN *******************************

INITTICK        LD      HL,TICLST       ; Zeiger auf die Tickerliste
                LD      DE,INTERRUPT    ; Zeiger auf Interruptroutine
                LD      B,#82           ; Ereignisklasse
                LD      C,#00           ; ROM-Auswahladresse
                CALL    KL_NEW_FAST_TIC ; Fast-Ticker initialisieren
                RET                     ; Ruecksprung nach Basic

; ************* DIE INTERRUPT-ROUTINE **********************************

INTERRUPT       DI                      ; Besser keine Interrupts
                PUSH    BC              ; BC sichern
                PUSH    DE              ; DE sichern
                PUSH    HL              ; HL sichern

                LD      A,(HOLD_VALUE)  ; Wert aus Speicherstelle laden
                INC     A               ; Plus 1
                LD      (HOLD_VALUE),A  ; Wieder speichern
                LD      (#C000),A       ; In den Bildschirmspeicher

                POP     HL              ; HL restaurieren
                POP     DE              ; DE restaurieren
                POP     BC              ; BC restaurieren
                EI                      ; Interrupts zulassen
                RET                     ; Ruecksprung zum IR-Handler
```

**Listing 1. Der Assemblercode zu »Flimmern«**

res als ein Event-Block, der um einen zwei Byte langen Verkettungszeiger erweitert ist:
Byte 0,1: Verkettungszeiger
Ab Byte 2: Normaler Event-Block

Der Ticker-Block weißt hingegen mehrere Unterschiede auf:
Byte 0,1: Verkettungszeiger
Byte 2,3: Countdown-Zähler
Byte 4,5: Wiederanlaufwert für den Zähler
Ab Byte 6: Normaler Event-Block

Das Betriebssystem des Schneider CPC kennt verschiedene Routinen, um die Interrupts zu verwalten. In der Tabelle finden Sie die gesamte Firmware, die sich mit diesem Thema beschäftigt. Die Startadresse und die Registerbelegung beim Aufruf und bei der Rückkehr stehen hier genauso wie eine kurze Erklärung der Aufgabe.

Nach diesem Ausflug in die theoretischen »Niederungen« des Firmware-Handbuchs sehen wir uns jetzt ein praktisches Beispiel an. Es demonstriert auf einfache Weise das Programmieren von Interrupts.

## Flimmern mit Interrupt

Für andere Aufgaben brauchen Sie nur die eigentliche Interruptroutine nach Ihren Anforderungen anzupassen. Das ganze »Beiwerk« können sie problemlos übernehmen.

Listing 1 ist das Assemblerlisting von »Flimmern«. In Listing 2 steht das Programm als Basic-Lader. Damit ist es auch unter Basic leicht einzugeben. Das Programm schreibt sich ständig ändernde Werte in das erste Byte des Bildschirmspeichers. Sie sehen damit in der linken oberen Ecke einen flimmernden Punkt. Und so sieht die Kernroutine des Interrupts aus:

```
LD  A,(HOLD_VALUE)
INC A
LD  (HOLD_VALUE),A
LD  (&C000),A
```

Vorher müssen allerdings alle wichtigen Register der CPU gesichert werden. Die Programmteile am Ende stellen den ursprünglichen Zustand wieder her.

## EI, DI – der Z80 spricht Kinderdialekt

Mit einer besonderen Tücke warten die Assemblerbefehle »DI« und »EI« auf. Falls ein Interrupt sehr häufig angefordert wird und die Routine selbst sehr zeitintensiv ist, dann läuft der Interruptpuffer über. Das Betriebssystem merkt diesen Fehler und blockt ihn auch ab. Dabei gehen aber einige Interrupts verloren. Verhindern läßt sich dieses Problem dadurch, daß zu Beginn des Unterprogramms alle Interrupts gesperrt und später wieder zugelassen werden. Doch Vorsicht bei solchen Programmen! Auch viele Firmwareroutinen geben die Interrupts automatisch wieder frei. Bei jedem Aufruf einer anderen Routine in dem selbstprogrammierten Programm muß diese auf Freigabe des Interrupts geprüft werden.

Unsere Routine benutzt trotzdem einen Fast-Ticker-Interrupt. Das Betriebssystem stellt dem Programmierer für den Fast-Ticker nämlich eine sehr komfortable Systemroutine, KL NEW FAST TICKER, zur Verfügung. Diese Firmwareroutine stellt den Ticker-Block zusammen.

Der normale Ticker besitzt keine mit KL NEW FAST TICKER vergleichbare Routine. In diesem Fall muß man sich deshalb den Ticker-Block selbst zusammenbasteln. In unserer Routine genügt es, beim Fast-Ticker unter dem Label TICLST Leerbytes anzugeben. Die Mischung der Pseudo-Assemblerdirektiven DEFB, DEFW und DEFS verdeutlicht die Struktur des Fast-Ticker-Blocks.

Label INITTICK ist die Startadresse der Assemblerroutine, die den Interrupt in die Interruptkette des Betriebssy-

```
; ********************************************
; *                                          *
; *  FNORMAL.ASM - Normale Ticker-Interrupts *
; *                                          *
; ********************************************

                ORG     &A000
                JP      INIT_EVENTBLOCK

KL_ADD_TICKER   EQU     #BCE9           ; KL ADD TICKER
KL_DEL_TICKER   EQU     #BCEC           ; KL DEL TICKER
KL_INIT_EVENT   EQU     #BCEF           ; KL INIT EVENT
HOLD_VALUE      DEFB    0

; ************* TICKERLISTE ********************************************

TICKERLIST      DEFW    0               ; Verkettungszeiger
                DEFW    0               ; Countdown-Zaehler
                DEFW    0               ; Wiederanlaufwert

EVENTBLOCK      DEFW    0               ; Kettungszeiger fuer Warteschlange
                DEFB    0               ; Ereigniszaehler
                DEFB    0               ; Ereignisklasse
                DEFW    0               ; Adresse der Interruptroutine
                DEFB    0               ; ROM-Auswahlwert

; ************* INTERRUPT INITIALISIEREN ********************************

INIT_EVENTBLOCK LD      HL,EVENTBLOCK   ; Zeiger auf den Event-Block
                LD      B,%10000000     ; Ereignisklasse
                LD      C,0             ; ROM-Auswahladresse
                LD      DE,INTERRUPT    ; Zeiger auf Interruptroutine
                CALL    KL_INIT_EVENT   ; Event-Block erstellen lassen

INIT_TICKERLIST LD      HL,TICKERLIST   ; Zeiger auf die Ticker-Liste
                LD      DE,1            ; Countdown-Zaehler
                LD      BC,1            ; Wiederanlaufwert
                CALL    KL_ADD_TICKER   ; Ticker initialisieren
                RET                     ; Ruecksprung nach Basic

; ************* DIE INTERRUPT-ROUTINE ***********************************

INTERRUPT       DI                      ; Besser keine Interrupts
                PUSH    BC              ; BC sichern
                PUSH    DE              ; DE sichern
                PUSH    HL              ; HL sichern

                LD      A,(HOLD_VALUE)  ; Wert aus Speicherstelle laden
                INC     A               ; Plus 1
                LD      (HOLD_VALUE),A  ; Wieder speichern
                LD      (#C000),A       ; In den Bildschirmspeicher

                POP     HL              ; HL restaurieren
                POP     DE              ; DE restaurieren
                POP     BC              ; BC restaurieren
                EI                      ; Interrupts zulassen
                RET                     ; Ruecksprung zum IR-Handler
```

**Listing 3. »Fnormal« als Assemblercode**

stems einfügt. Dazu benötigt sie folgende Daten in den Registern des Z80-Prozessors:

HL – Zeiger auf die Fast-Ticker-Liste
DE – Zeiger auf die Interruptroutine
C – ROM-Auswahladresse (ROM-Select)
B – Ereignisklasse

Der Wert 82hex (130dez) im B-Register ist Bit für Bit organisiert. Er läßt sich folgendermaßen zerlegen:

Bitnummer : 76543210
82hex binär: 10000010

Dabei haben die Bits folgende Bedeutung:

Bit 0: Interrupt steht im RAM
Bit 1-4: Priorität (ist nur bei synchronen Ereignissen von Bedeutung
Bit 5 : immer Null
Bit 6 : kein eiliger Interrupt
Bit 7 : asynchroner Interrupt

Das Programm wird nach dem Assemblieren mit

```
CALL &A000
```

aufgerufen. Irgendwo auf dem Bildschirm sehen Sie jetzt ein nervöses Flimmern. Finden Sie es nicht, so geben Sie

```
MODE 2
```

ein. Dies löst eine Reorganisation des Bildschirmspeichers aus.

Der Speicherbereich ab A000hex, in dem der Fast-Ticker-Block und die Maschinenroutine abgelegt sind, ist ab sofort tabu. POKEn von Werten oder erneutes Assemblieren des Quelltextes auf derselben Adresse führt fast immer zum Systemabsturz.

Bevor Sie die im folgenden beschriebenen Änderungen vornehmen, müssen Sie stets den Computer mit <CTRL+SHIFT+ESC> oder

```
CALL 0
```

zurücksetzen. Erst dann löscht das Betriebssystem auch die Interruptroutine aus der Fast-Ticker-Kette. Sie können sich aber auch eine Maschinencode-Routine schreiben, die den Fast-Ticker-Interrupt aus der Liste streicht:

```
DEL_FAST_TICKER: EQU  &BCE6

DEL_FTICKER:     LD   HL,TICLST
                 CALL DEL_FAST_TICKER
                 RET
```

Bei den Routinen KL DEL FAST TICKER (BCE6hex), KL DEL FRAME FLY (BCDDhex, wird hier nicht behandelt) und KL DEL TICKER (BCEChex) ist folgendes zu beachten. Deren Aufruf sorgt dafür, daß ab diesem Zeitpunkt keine Interrupt-Anforderungen mehr beachtet werden. Bereits erkannte Anforderungen, die lediglich noch nicht bearbeitet wurden, werden jedoch noch ausgeführt. Wollen Sie dies verhindern, sperren Sie einfach mit KL DEL SYNCHRONOUS (Adresse BCF8hex) beziehungsweise KL DISARM EVENT (BD0Ahex) den Aufruf. Dabei ist die

```
; ************************************************
; *                                              *
; *    Drucker-Spooler fuer den Schneider-CPC    *
; *                                              *
; ************************************************

                ORG     &A000

                JP      INIT_RSX

; ************* EQUATES FUER DEN SPOOLER ********************************

TXT_OUTPUT      EQU     #BB5A   ; TXT OUTPUT
KL_LOG_EXT      EQU     #BCD1   ; KL LOG EXT
KL_NEW_FAST_TIC EQU     #BCE0   ; KL NEW FAST TICKER
KL_DEL_FAST_TIC EQU     #BCE6   ; KL DEL FAST TICKER
MC_PRINT_CHAR   EQU     #BD2B   ; MC PRINT CHAR
MC_BUSY_PRINTER EQU     #BD2E   ; MC BUSY PRINTER
IND_WAIT_CHAR   EQU     #BDF1   ; IND MC WAIT CHAR

; ************* SPEICHERSTELLEN *****************************************

SPSTART         DEFS    2       ; Startadresse des Spool-Speichers
SPLAST          DEFS    2       ; Endadresse des Spool-Speichers
SPPNTR          DEFS    2       ; Belegungszeiger des Spool-Speichers
IRPNTR          DEFS    2       ; "Ausgedruckt"-Zeiger des Spool-Speichers
EMPTY           DEFS    1       ; Flag, ob Spool-Speicher leer ist
PRINTCHAR       DEFS    1       ; Speicherstelle fuer das Druckzeichen
WAIT_CHAR_BUF   DEFS    3       ; Speicher fuer alten Druckervektor
; ************* TICKER-LISTE FUER DEN SPOOLER ***************************

TICLST          DEFW    0,0
                DEFB    #00,#83
                DEFW    SP_INTERRUPT
                DEFS    2

; ************* DATEN ZUR EINBINDUNG DER RSX-BEFEHL *********************

RSX_TABLE       DEFW    NAMES           ; Zeiger auf Namenstabelle
                JP      SPOOL           ; Sprung auf SPOOL
                JP      UNSPOOL         ; Sprung auf UNSPOOL

NAMES           DEFM    "SPOO"          ; Befehlsname "SPOOL"
                DEFB    #CC
                DEFM    "UNSPOO"        ; Befehlsname "UNSPOOL"
                DEFB    #CC
                DEFB    #00             ; Ende der Tabelle

SPACE           DEFS    #04             ; Hilfsspeicher

; ************* EINBINDUNG DES SPOOLERS ALS RSX-BEFEHLE *****************

INIT_RSX LD     BC,RSX_TABLE    ; Zeiger auf RSX-Sprungtabelle
         LD     HL,SPACE        ; Zeiger auf Hilfsspeicher
         CALL   KL_LOG_EXT      ; RSXen initialisieren
         RET                    ; Ruecksprung nach Basic

; ************* RSX "SPOOL" - SPOOLER EINSCHALTEN ***********************

SPOOL    CP     2               ; Werden zwei Parameter uebergeben?
         JR     NZ,ERROR        ; Nein - Fehler!

GETADDR  LD     H,(IX+3)        ; \ Startadresse des Spool-Speichers
         LD     L,(IX+2)        ; / aus dem RSX-Aufruf holen
         LD     (SPSTART),HL    ; Und speichern
         LD     (SPPNTR),HL     ; dito, der Speicher ist noch leer
         LD     (IRPNTR),HL     ; dito, wir haben noch nichts gedruckt

GETLENG  LD     D,(IX+1)        ; \ Laenge des Spool-Speichers
         LD     E,(IX+0)        ; / aus dem RSX-Aufruf holen
         ADD    HL,DE           ; Und zur Startadresse addieren
         LD     (SPLAST),HL     ; Als Endadresse abspeichern

SETEMPTY SUB    A               ; Akku loeschen
         LD     (EMPTY),A       ; Als Flag nach EMPTY schreiben

COPY     LD     HL,IND_WAIT_CHAR ; Zeiger auf die Indirection
         LD     DE,WAIT_CHAR_BUF ; Zeiger auf den Pufferspeicher
         LD     BC,3            ; 3 Byte bei LDIR zu kopieren
         LDIR                   ; Und kopieren

REPLACE  LD     HL,NEW_WAIT_CHAR     ; Adresse des neuen Druckervektors
         LD     (IND_WAIT_CHAR+1),HL ; In den Originalvektor eintragen
```

```
INITTICK LD      HL,TICLST          ; Zeiger auf die Ticker-Liste
         LD      DE,SP_INTERRUPT    ; Zeiger auf Interruptroutine
         LD      B,#83              ; Ereignisklasse
         LD      C,#00              ; ROM-Auswahladresse
         CALL    KL_NEW_FAST_TIC    ; Fast-Ticker initialisieren
         RET                        ; Ruecksprung nach Basic

; ************* RSX "UNSPOOL" - SPOOLER ABSCHALTEN **********************

UNSPOOL  LD      HL,WAIT_CHAR_BUF   ; Vektor fuer die Druckerausgabe aus
         LD      DE,IND_WAIT_CHAR   ; dem Puffer "WAIT_CHAR_BUF" wieder
         LD      BC,3               ; mit LDIR zurueckkopieren
         LDIR

RESTORE  LD      HL,TICLST          ; Zeiger auf Ticker-Liste
         CALL    KL_DEL_FAST_TIC    ; Fast-Ticker deinstallieren
         RET                        ; Ruecksprung nach Basic

; ************* FEHLER-ROUTINE ******************************************

ERROR    LD      A,7                ; Code fuer BEEP
         CALL    TXT_OUTPUT         ; Ausgeben
         RET                        ; Ruecksprung nach Basic

; ************* VERGLEICHE HL- UND DE-REGISTER **************************

CP_HL_DE PUSH    HL
         OR      A
         SBC     HL,DE
         POP     HL
         RET
; ************* NEUE ROUTINE ZUR ZEICHENAUSGABE *************************

NEW_WAIT_CHAR    LD      (PRINTCHAR),A
                 PUSH    BC
                 PUSH    DE
                 PUSH    HL
                 DI
                 LD      HL,(SPPNTR)
                 LD      DE,(SPLAST)
                 CALL    CP_HL_DE
                 JR      C,SP_OKCHAR

SP_MEMFULL       LD      A,(EMPTY)
                 CP      255
                 JR      Z,SP_CONT
                 LD      HL,(SPSTART)
                 LD      (SPPNTR),HL
                 JR      SP_OKCHAR

SP_CONT          EI
                 SCF                        ; \ Setzt das Carry-Flag
                 CCF                        ; / zurueck - Z80-Trick!
                 POP     HL
                 POP     DE
                 POP     BC
                 LD      A,(PRINTCHAR)
                 RET

SP_OKCHAR        LD      A,(PRINTCHAR)
                 LD      (HL),A
                 INC     HL
                 LD      (SPPNTR),HL
                 POP     HL
                 POP     DE
                 POP     BC
                 LD      A,255
                 LD      (EMPTY),A
                 EI
                 SCF
                 RET

; ************* ROUTINE, DIE DIE ZEICHEN PER INTERRUPT AUSGIBT ************

SP_INTERRUPT     DI
                 PUSH    BC
                 PUSH    DE
                 PUSH    HL
                 CALL    MC_BUSY_PRINTER
                 JR      C,BUSY
                 DI
                 PUSH    AF
                 LD      A,(EMPTY)
```

**Listing 5. Ein Druckerspooler ist schnell eingerichtet**

Routine an BCF8hex für synchrone Ereignisse, die Systemroutine KL DISARM EVENT für asynchrone Events vorgesehen.

In unserem Programm haben wir uns für einen asynchronen Interrupt entschieden. Weshalb ist hier ein synchroner Interrupt wohl nicht sinnvoll?

Ersetzen Sie

`LD B,#82`

durch

`LD B,#02`

Nun ist im B-Register das 7. Bit zurückgesetzt. Das sagt dem Betriebssystem, daß ein synchroner Interrupt initialisiert wird.

Sobald Sie die Routine mit

`CALL &A000`

initialisieren, entdecken Sie, daß der Interrupt nicht beachtet wird. Wie kommt das?

Synchrone Ereignisse werden per Polling erfaßt, also durch regelmäßiges Nachschauen, ob ein Interrupt angefordert wird. Solange sich aber der Basic-Interpreter im Direktmodus befindet, führt er das Polling nicht durch.

Geben Sie nun

`FOR i=1 TO 10000:NEXT i`

ohne Zeilennummer ein. Während dem Bearbeiten dieser leeren FOR-NEXT-Schleife sehen Sie wieder das vertraute Flimmern auf dem Bildschirm. Während das Locomotive-Basic Befehle interpretiert, ruft es regelmäßig auch die Pollingroutine im Basic-ROM auf.

`10 GOTO 10`

erfüllt übrigens denselben Zweck.

## Normale Ticker-Ereignisse

In den meisten Fällen braucht man den Fast-Ticker gar nicht. Der normale Ticker, der im 1/50-Sekunden-Takt arbeitet, reicht meist aus. Er hat sogar den Vorteil, daß man das Zeitintervall – ähnlich dem Basic-Befehl EVERY – bestimmen kann.

Listing 3 (der Basic-Lader davon steht in Listing 4) zeigt den Assemblercode von »Fnormal«. Es ist eine geänderte Version des Flimmern und macht prinzipiell das gleiche. Fnormal wird nur vom Normal-Ticker gesteuert.

Die Einreihung von Interrupts in die normale Ticker-Kette ist etwas aufwendiger als beim Fast-Ticker. Das Rahmenprogramm können Sie wieder für eigene Routinen übernehmen. Die Datenstruktur, die dem Betriebssystem zu übergeben ist, setzt sich aus einem normalen Event-Block und den vorangestellten Bytes für den Ticker zusammen:

Byte 0,1: Verkettungszeiger
Byte 2,3: Countdown-Zähler

Byte 4,5: Wiederanlaufwert für den Zähler
Byte 6,7: Zeiger auf die Kette der Warteschlange
Byte 8: Ereigniszähler
Byte 9: Ereignisklasse
Byte 10,11: Adresse der Interruptroutine
Byte 12: ROM-Auswahladresse

Mit dem 6. Byte der Liste beginnt damit wieder der normale Event-Block.

Ein Ticker-Interrupt wird in zwei Arbeitsgängen initialisiert: dem Einbinden des Event-Blocks und des (gesamten) Ticker-Blocks.

Zum Lösen der ersten Aufgabe gibt es zwei Wege. Der umständlichere – dafür speicherplatzsparende – bedeutet, den Event-Block »von Hand« mit DEFB und DEFW anzugeben. Da die Größe des RAM-Bereichs des Schneiders für Maschinenprogramme fast ausreicht, wollen wir aber komfortabler arbeiten. Das Stichwort dazu heißt: KL INIT EVENT.

KL INIT EVENT initialisiert den Ereignis- oder Event-Block. Dazu braucht die Routine folgende Werte in den angegebenen Registern:

HL – zeigt auf den Ereignisblock
B – enthält die Ereignisklasse
C – enthält das ROM-Select-Byte
DE – zeigt auf die Ereignisroutine.

Die Bits im B-Register bedeuten:

Bit 0=0: Die Routine liegt im zentralen RAM zwischen den Adressen 4000 und BFFFhex
Bit 6=0: Es ist kein eiliges Ereignis
Bit 7=0: Es handelt sich um ein asynchrones Ereignis.

Da die Routine als asynchroner Event eingebunden ist, sind die Prioritätsbits 1 bis 4 ohne Bedeutung. KL INIT EVENT legt die Werte aus den Registern in der richtigen Reihenfolge im Event-Block ab.

Der Programmteil INIT__EVENTBLOCK ruft KL INIT EVENT automatisch auf. INIT__TICKERLIST ruft das Betriebssystem mit Hilfe der Firmwareroutine KL ADD TICKER und baut damit die Ticker-Kette neu auf. Dazu erwartet diese im HL-Register die Adresse des Ticker-Blocks. In DE wird der Ausgangswert des Countdown-Zählers übergeben und in BC der »Wiederanlaufwert«.

Normale Ticker-Interrupts müssen nicht zwangsweise 50mal in der Sekunde aufgerufen werden. Die geeignete Wahl der Inhalte für die Register DE und BC läßt auch andere Zeiteinheiten zu. Der Countdown-Zähler bestimmt dabei, nach wieviel 1/50 Sekunden die Routine zum ersten Mal aufgerufen wird. Der Wiederanlaufwert gibt an, nach welchem Zeitintervall der zweite und alle folgenden Aufrufe ange-

```
                CP      255
                JR      NZ,BUSY2
                POP     AF
                LD      HL,(IRPNTR)
                LD      DE,(SPLAST)
                CALL    CP_HL_DE
                JR      NC,BUSY3
                LD.     DE,(SPPNTR)
                CALL    CP_HL_DE
                JR      NC,SP_EMPTYAGN
                LD      A,(HL)
                INC     HL
                LD      (IRPNTR),HL
                POP     HL
                POP     DE
                POP     BC
                EI
                CALL    WAIT_CHAR_BUF
                RET

SP_EMPTYAGN     LD      HL,(SPSTART)
                LD      (SPPNTR),HL
                LD      (IRPNTR),HL
                SUB     A
                LD      (EMPTY),A
                JR      BUSY

BUSY3           LD      A,(EMPTY)
                CP      0
                JR      Z,BUSY
                JR      SP_EMPTYAGN

BUSY2           POP     AF

BUSY            POP     HL
                POP     DE
                POP     BC
                EI
                RET
```

**Listing 5. Ein Druckerspooler ist schnell eingerichtet (Schluß)**

```
; ************************************
; *                                  *
; *  CURBLINK.ASM - Blinkender Cursor *
; *                                  *
; ************************************

                ORG     &A000
                JP      INIT_EVENTBLOCK

KL_ADD_TICKER   EQU     #BCE9           ; KL ADD TICKER
KL_DEL_TICKER   EQU     #BCEC           ; KL DEL TICKER
KL_INIT_EVENT   EQU     #BCEF           ; KL INIT EVENT
TXT_CUR_ENABLE  EQU     #BB7B           ; TXT CUR ENABLE
TXT_CUR_DISABLE EQU     #BB7E           ; TXT CUR DISABLE

HOLD_VALUE      DEFB    0

; ************* TICKERLISTE *********************************************

TICKERLIST      DEFW    0               ; Verkettungszeiger
                DEFW    0               ; Countdown-Zaehler
                DEFW    0               ; Wiederanlaufwert

EVENTBLOCK      DEFW    0               ; Kettungszeiger fuer Warteschlange
                DEFB    0               ; Ereigniszaehler
                DEFB    0               ; Ereignisklasse
                DEFW    0               ; Adresse der Interruptroutine
                DEFB    0               ; ROM-Auswahlwert

; ************* INTERRUPT INITIALISIEREN ********************************

INIT_EVENTBLOCK LD      HL,EVENTBLOCK   ; Zeiger auf den Event-Block
                LD      B,%10000000     ; Ereignisklasse
                LD      C,0             ; ROM-Auswahladresse
                LD      DE,INTERRUPT    ; Zeiger auf Interruptroutine
                CALL    KL_INIT_EVENT   ; Event-Block erstellen lassen

INIT_TICKERLIST LD      HL,TICKERLIST   ; Zeiger auf die Ticker-Liste
                LD      DE,20           ; Countdown-Zaehler
                LD      BC,20           ; Wiederanlaufwert
                CALL    KL_ADD_TICKER   ; Ticker initialisieren
                RET                     ; Ruecksprung nach Basic
```

```
; ************* DIE INTERRUPT-ROUTINE ************************************

INTERRUPT       DI                                  ; Besser keine Interrupts
                PUSH    BC                          ; BC sichern
                PUSH    DE                          ; DE sichern
                PUSH    HL                          ; HL sichern

CHECK_CURSOR    LD      A,(HOLD_VALUE)              ; Speicherstelle untersuchen
                CP      0                           ; Wenn 0 - Cursor einschalten
                JR      Z,CURSOR_ON                 ; Sonst ausschalten

CURSOR_OFF      CALL    TXT_CUR_DISABLE             ; Schaltet den Cursor ab
                SUB     A                           ; Und laedt den entgegengesetzten
                LD      (HOLD_VALUE),A              ; Wert in die Speicherstelle
                JR      END_INTERRUPT

CURSOR_ON       CALL    TXT_CUR_ENABLE              ; Schaltet den Cursor ein
                LD      A,255                       ; Und laedt den entgegengesetzten
                LD      (HOLD_VALUE),A              ; Wert in die Speicherstelle

END_INTERRUPT   POP     HL                          ; HL restaurieren
                POP     DE                          ; DE restaurieren
                POP     BC                          ; BC restaurieren
                EI                                  ; Interrupts zulassen
                RET                                 ; Ruecksprung zum IR-Handler
```

**Listing 7. Der Assemblercode zum Einbinden des blinkenden Cursors in das Betriebssystem**

fordert werden. Mit dem minimalen Wert 1, sowohl für das BC- und das DE-Register, wird die Interruptroutine wirklich alle $^{1}/_{50}$-Sekunden abgearbeitet. Geben Sie in BC den Wert 10 und in DE 5 an, wird die Interruptroutine erstmals nach $^{1}/_{5}$ Sekunde und im folgenden alle $^{1}/_{10}$ Sekunden aufgerufen.

Ändern Sie im Listing 3 die nach dem Label INIT__TICKERLIST stehenden Befehle in

```
LD    HL,TICKERLIST
LD    DE,50
LD    BC,50
CALL KL_ADD_TICKER
RET
```

ab. Als Ergebnis sehen Sie eine etwas ungewöhnlich aussehende »Uhr«, die die Sekunden durch gesetzte und gelöschte Punkte anzeigt.

Wird der Routine KL ADD TICKER ein Countdown-Wert von Null übergeben, reiht sie den Interrupt zwar in die Ticker-Kette ein, arbeitet ihn aber nie ab. Er ist inaktiv. Ist der Wiederanlaufwert Null,

```
100 ' FLIMMER.ASM als Basic-Ladeprogramm
                                           [EBFA]
110 ' ---------------------------------
                                           [27E6]
120 MEMORY 40959                           [706E]
130 FOR i=40960 TO 41006                   [CD84]
140 READ a:POKE i,a:NEXT i                 [07A6]
150 MODE 2:PRINT:PRINT:CALL &A000          [2A50]
160 END                                    [EF1C]
170 ' Maschinenprogramm in DATA-Zeilen     [497E]
180 ' -------------------------------      [CF40]
190 DATA &C3,&0E,&A0,&00,&00,&00,&00,&00
    ,&00,&00,&00,&00,&00,&00,&21,&04       [3656]
200 DATA &A0,&11,&1C,&A0,&06,&82,&0E,&00
    ,&CD,&E0,&BC,&C9,&F3,&C5,&D5,&E5       [C43C]
210 DATA &3A,&03,&A0,&3C,&32,&03,&A0,&32
    ,&00,&C0,&E1,&D1,&C1,&FB,&C9           [1542]
```

**Listing 2. »Flimmern« als Basic-Lader**

```
100 ' FNORMAL.ASM - Basic-Ladeprogramm     [61DA]
110 ' -------------------------------      [5F32]
120 MEMORY 40959                           [706E]
130 FOR i=40960 TO 41021                   [CD7E]
140 READ a:POKE i,a:NEXT i                 [07A6]
150 MODE 2                                 [A85A]
160 CALL &A000:PRINT:PRINT                 [41F0]
170 END                                    [981E]
180 ' Maschinenprogramm in DATA-Zeilen     [3D80]
190 ' -------------------------------      [8F42]
200 DATA &C3,&11,&A0,&00,&00,&00,&00,&00
    ,&00,&00,&00,&00,&00,&00,&00,&00       [A412]
210 DATA &00,&21,&0A,&A0,&06,&80,&0E,&00
    ,&11,&2B,&A0,&CD,&EF,&BC,&21,&04       [A39C]
220 DATA &A0,&11,&01,&00,&01,&01,&00,&CD
    ,&E9,&BC,&C9,&F3,&C5,&D5,&E5,&3A       [26EC]
230 DATA &03,&A0,&3C,&32,&03,&A0,&32,&00
    ,&C0,&E1,&D1,&C1,&FB,&C9               [AABA]
```

**Listing 4. »Fnormal« ist nicht so gut wie »Flimmern«**

```
100 ' CURBLINK.ASM - Basic-Ladeprogramm    [E970]
110 ' --------------------------------     [008C]
120 MEMORY 40959                           [706E]
130 FOR i=40960 TO 41035                   [CE88]
140 READ a:POKE i,a:NEXT i                 [07A6]
150 CALL &A000:END                         [C8F4]
160 ' Maschinenprogramm in DATA-Zeilen     [957C]
170 ' -------------------------------      [4F3E]
180 DATA &C3,&11,&A0,&00,&00,&00,&00,&00
    ,&00,&00,&00,&00,&00,&00,&00,&00       [4320]
190 DATA &00,&21,&0A,&A0,&06,&80,&0E,&00
    ,&11,&2B,&A0,&CD,&EF,&BC,&21,&04       [4EAA]
200 DATA &A0,&11,&14,&00,&01,&14,&00,&CD
    ,&E9,&BC,&C9,&F3,&C5,&D5,&E5,&3A       [F8F8]
210 DATA &03,&A0,&FE,&00,&28,&09,&CD,&7E
    ,&BB,&97,&32,&03,&A0,&18,&08,&CD       [FAF6]
220 DATA &7B,&BB,&3E,&FF,&32,&03,&A0,&E1
    ,&D1,&C1,&FB,&C9                       [906E]
```

**Listing 8. Der Basic-Lader für den blinkenden Cursor**

```
100 ' SPOOL.ASM - Basic-Ladeprogramm       [67D6]
110 ' ------------------------------       [967E]
120 MEMORY 40959                           [706E]
130 FOR i=40960 TO 41260                   [DB88]
140 READ a:POKE i,a:NEXT i                 [07A6]
150 CALL &A000 ' Spooler initialisieren    [D35A]
160 PRINT                                  [FB88]
170 PRINT "Befehle:<2>|SPOOL,Startadress
    e,Laenge"                              [4FE0]
180 PRINT "<10>|UNSPOOL"                   [FA2C]
190 PRINT:PRINT                            [CD1C]
200 END                                    [A912]
210 ' Maschinenprogramm in DATA-Zeilen     [C074]
220 ' -------------------------------      [5236]
230 DATA &C3,&33,&A0,&00,&00,&00,&00,&00
    ,&00,&00,&00,&00,&00,&00,&00,&00       [9820]
240 DATA &00,&00,&00,&00,&00,&83,&DA,&A0
    ,&00,&00,&22,&A0,&C3,&3D,&A0,&C3       [3E1C]
250 DATA &7D,&A0,&53,&50,&4F,&4F,&CC,&55
    ,&4E,&53,&50,&4F,&4F,&CC,&00,&00       [7A04]
260 DATA &00,&00,&00,&01,&1A,&A0,&21,&2F
    ,&A0,&CD,&D1,&BC,&C9,&FE,&02,&20       [1DC4]
270 DATA &4E,&DD,&66,&03,&DD,&6E,&02,&22
    ,&03,&A0,&22,&07,&A0,&22,&09,&A0       [839C]
280 DATA &DD,&56,&01,&DD,&5E,&00,&19,&22
    ,&05,&A0,&97,&32,&0B,&A0,&21,&F1       [35A8]
290 DATA &BD,&11,&0D,&A0,&01,&03,&00,&ED
    ,&B0,&21,&9B,&A0,&22,&F2,&BD,&21       [37D2]
300 DATA &10,&A0,&11,&DA,&A0,&06,&83,&0E
    ,&00,&CD,&E0,&BC,&C9,&21,&0D,&A0       [FDEC]
310 DATA &11,&F1,&BD,&01,&03,&00,&ED,&B0
    ,&21,&10,&A0,&CD,&E6,&BC,&C9,&3E       [A220]
320 DATA &07,&CD,&5A,&BB,&C9,&E5,&B7,&ED
    ,&52,&E1,&C9,&32,&0C,&A0,&C5,&D5       [1FAC]
330 DATA &E5,&F3,&2A,&07,&A0,&ED,&5B,&05
    ,&A0,&CD,&95,&A0,&38,&19,&3A,&0B       [EB32]
340 DATA &A0,&FE,&FF,&28,&08,&2A,&03,&A0
    ,&22,&07,&A0,&18,&0A,&FB,&37,&3F       [B40E]
350 DATA &E1,&D1,&C1,&3A,&0C,&A0,&C9,&3A
    ,&0C,&A0,&77,&23,&22,&07,&A0,&E1       [BFEC]
360 DATA &D1,&C1,&3E,&FF,&32,&0B,&A0,&FB
    ,&37,&C9,&F3,&C5,&D5,&E5,&CD,&2E       [F7D8]
370 DATA &BD,&38,&45,&F3,&F5,&3A,&0B,&A0
    ,&FE,&FF,&20,&3B,&F1,&2A,&09,&A0       [977E]
380 DATA &ED,&5B,&05,&A0,&CD,&95,&A0,&30
    ,&25,&ED,&5B,&07,&A0,&CD,&95,&A0       [815A]
390 DATA &30,&0D,&7E,&23,&22,&09,&A0,&E1
    ,&D1,&C1,&FB,&CD,&0D,&A0,&C9,&2A       [BC44]
400 DATA &03,&A0,&22,&07,&A0,&22,&09,&A0
    ,&97,&32,&0B,&A0,&18,&0A,&3A,&0B       [E250]
410 DATA &A0,&FE,&00,&28,&03,&18,&E8,&F1
    ,&E1,&D1,&C1,&FB,&C9                   [92AC]
```

**Listing 6. Der Basic-Lader zu »Spool«**

dann beachtet das System die Routine genau einmal und erklärt sie dann für inaktiv.

Solche inaktiven Interrupts haben in der Ticker-Kette keine Aufgabe, kosten aber wertvolle Rechenzeit. Zum Löschen von normalen Ticker-Interrupts dient die Routine KL DEL TICKER:

```
KL_DEL_TICKER EQU  &BCEC

DELETE_TICKER LD   HL,TICKERLIST
              CALL KL_DEL_TICKER
              RET
```

Damit kennen Sie fast alles, was zur Interrupt-Programmierung wissenswert ist. Zum Abschluß zeigen wir noch zwei Programme, die den Interrupt sinnvoll ausnutzen.

Listing 5 ist der Assemblercode für einen kompletten Druckerspooler. Die Aufgabe eines Druckerspoolers besteht darin, alle Zeichen, die der Computer an den Drucker schickt, intern zwischenzuspeichern. Der Spooler gibt die Zeichen dann mittels Interrupt an den Drucker weiter. Dadurch arbeitet der Computer schon längst weiter, während der (langsamere) Drucker noch beschäftigt ist. Listing 6 zeigt das gleiche Programm als Basic-Lader.

Bei »Spool« handelt es sich nun aber nicht nur um einen »nackten« Spooler, sondern um eine komfortable Basic-Erweiterung als RSX-Befehl. Nachdem der Spooler mit

`CALL &A000`

initialisiert ist, kennt Ihr Computer zwei neue Befehle:

`|SPOOL,Startadresse,Länge`

`|UNSPOOL`

Der erste Befehl initialisiert den Spooler. Dabei verändert das Programm den Systemvektor zur Druckerausgabe und initialisiert einen Fast-Ticker-Interrupt. Sie können dazu einen beliebigen Speicherbereich festlegen. Brauchen Sie beispielsweise einen 10000 Zeichen großen Puffer ab der Adresse 30000, so teilen Sie das dem Computer mit folgendem Befehl mit:

`MEMORY 29999:|SPOOL,30000,10000`

Normalerweise reicht ein Speicherbereich von drei bis vier KByte aus. Dieses »Speicherplatzopfer« jedoch lohnt sich. Mit

`|UNSPOOL`

wird der Spooler wieder abgeschaltet. Dabei dürfen aber keine Zeichen mehr im Zwischenspeicher stehen, da diese sonst unwiderruflich verlorengehen und nicht gedruckt werden.

Das Programm aus Listing 7 (Basic-Lader in Listing 8) simuliert einen blinkenden Cursor auf dem Bildschirm. Der von Schneider serienmäßig benutzte stehende Cursor ist oft auf dem Bildschirm etwas schwierig zu finden. Ein blinkender Cursor fällt hingegen sofort ins Auge. Mit »Curblink« erweitern Sie Ihr Betriebssystem um dieses nützliche Utility.

Der Aufwand für das Programm ist nicht so groß. Es werden lediglich abwechselnd die Systemroutinen TXT CUR ENABLE und TXT CUR DISABLE aufgerufen. Diese ändern jeweils die Darstellung des Cursors. Als Countdown- und Wiederanlaufwert benutzt das Programm den Wert 20. Vermindern Sie die beiden Werte, so blinkt der Cursor schneller – vergrößern Sie ihn, so blinkt er langsamer.

(Martin Kotulla/hg)

---

## Frame-Flyback-Interrupt:

**BCD7hex – KL NEW FRAME FLY**
Initialisiert einen Block und übergibt ihn an die Frame-Flyback-Kette.
Einsprungbedingungen:
HL-Zeiger auf den Block.
B enthält die Ereignisklasse.
C enthält das ROM-Select-Byte.
DE zeigt auf die Ereignisroutine.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**BCDAhex – KL ADD FRAME FLY**
Übergibt einen bereits initialisierten Block an die Frame-Flyback-Chain.
Einsprungbedingungen:
HL zeigt auf den Block.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**BCDDhex – KL DEL FRAME FLY**
Löscht Block aus der Frame-Flyback-Chain.
Einsprungbedingungen:
HL zeigt auf den Block.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**Fast-Ticker-Interrupt:**

**BCE0hex – KL NEW FAST TICKER**
Initialisiert einen Block und übergibt ihn an die Fast-Ticker-Chain.
Einsprungbedingungen:
HL zeigt auf den Fast-Ticker-Block.
B enthält die Ereignisklasse.
C enthält das ROM-Select-Byte.
DE zeigt auf die Ereignisroutine.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**BCE3hex – KL ADD FAST TICKER**
Übergibt einen bereits initialisierten Block an die Fast-Ticker-Chain.
Einsprungbedingungen:
HL zeigt auf den Fast-Ticker-Block.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**BCE6hex – KL DEL FAST TICKER**
Löscht einen Block aus der Fast-Ticker-Chain.
Einsprungbedingungen:
HL zeigt auf den Fast-Ticker-Block.
Aussprungbedingungen:
AF, DE und HL sind zerstört.

**Normale Ticker-Interrupts:**

**BCE9hex – KL ADD TICKER**
Übergibt einen vorbereiteten Block an die normale Ticker-Chain.
Einsprungbedingungen:
HL zeigt auf den Ticker-Block.
DE enthält den Ausgangswert des Zählers.
BC enthält den Wiederanlaufwert des Zählers.
Aussprungbedingungen:
AF, BC und DE sind zerstört.

**BCEChex – KL DEL TICKER**
Löscht Block aus der normalen Ticker-Kette
Einsprungbedingungen:
HL zeigt auf den Ticker-Block.
Aussprungbedingungen:
Akku und HL sind zerstört.
Wenn der Ticker-Block gefunden wurde, ist das Carry-Flag eingeschaltet und DE enthält den Zähler vor dem nächsten Event. Wurde der Ticker-Block nicht in der Ticker-Chain aufgefunden, ist das Carry-Flag ausgeschaltet und der Inhalt des DE-Registers bedeutungslos.

**Ereignisse:**

**BCEFhex – KL INIT EVENT**
Initialisiert einen Event-Block aus den in Registern übergebenen Daten.
Einsprungbedingungen:
HL zeigt auf den Event-Block.
B enthält die Ereignisklasse.
C enthält die ROM-Auswahladresse.
DE zeigt auf die Event-Routine.
Aussprungbedingungen:
HL enthält die Adresse des Event-Blocks plus 7 und zeigt deshalb auf die erste Adresse hinter dem Event-Block.

**BCF2hex – KL EVENT**
Fordert die Bearbeitung einer Event-Routine an.
Einsprungbedingungen:
HL zeigt auf den Event-Block.
Aussprungbedingungen:
AF, BC, DE und HL sind zerstört.

**BCF5hex – KL SYNC RESET**
Löscht die Warteschlange für synchrone Ereignisse.
Einsprungbedingungen:
Es sind keine Registerwerte nötig.
Aussprungbedingungen:
AF und HL sind zerstört.

**BCF8hex – KL DEL SYNCHRONOUS**
Löscht ein synchrones Ereignis aus der Warteschlange und verhindert die Bearbeitung noch ausstehender Interrupt-Anforderungen.
Einsprungbedingungen:
HL zeigt auf den Event-Block.
Aussprungbedingungen:
AF, BC, DE und HL sind zerstört.

**BD04hex – KL EVENT DISABLE**
Sperrt die Bearbeitung normaler synchroner Ereignisse, so daß Interrupt-Anforderungen dem Hauptprogramm verborgen bleiben.
Einsprungbedingungen:
Keine Registerwerte nötig.
Aussprungbedingungen:
HL ist zerstört.

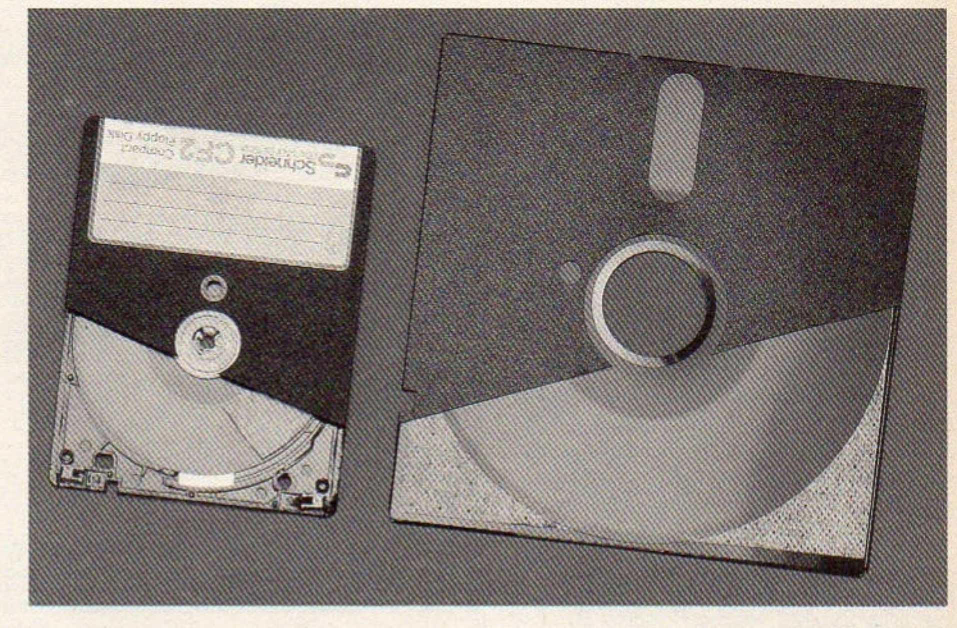
**BD07hex – KL EVENT ENABLE**
Ermöglicht wieder die Bearbeitung normaler synchroner Ereignisse.
Einsprungbedingungen:
Keine Registerwerte nötig.
Aussprungbedingungen:
HL ist zerstört.

**BD0Ahex – KL DISARM EVENT**
Verhindert das Auftreten eines Events. KL DISARM EVENT darf nur bei asynchronen Ereignissen benutzt werden.
Einsprungbedingungen:
HL zeigt auf den Event-Block.
Aussprungbedingungen:
Akku und Flag sind zerstört.

**Tabelle. Diese Firmwareroutinen brauchen Sie für Interruptsteuerung**

# Der Floppy aufs Bit geschaut

**Fast jeder Computerbesitzer hat ein Diskettenlaufwerk. Wir zeigen Ihnen, wie Sie ganz ohne Kenntnis von Maschinensprache Ihr Laufwerk voll ausreizen.**




Jeder Disketten-Controller für den Schneider CPC besteht im Prinzip nur aus zwei für den Programmierer interessanten Baugruppen. Da ist zum ersten der Ein-/Ausschalter für die Laufwerksmotoren, der über die Portadresse FA7E hex gesteuert wird. Durch den Befehl

```
OUT &FA7E,&01
```

werden die Motoren aller Laufwerke ein- und mit

```
OUT &FA7E,&00
```

wieder ausgeschaltet. Damit klärt sich auch die oft gestellte Frage, ob es korrekt ist, daß bei einem Zugriff auf eine Diskette die Motoren aller Laufwerke anlaufen.

Das Herz der zweiten Baugruppe ist das IC mit dem Namen µPD765 – der eigentliche Disketten-Controller. Dieser Baustein gibt sich dem Programmierer durch zwei Register, über die die Daten zwischen Computer und Laufwerk ausgetauscht werden, zu erkennen. Mit

```
PRINT INP(&FB7E)
```

wird der gerade aktuelle Zustand des Controllers und der angeschlossenen Laufwerke abgefragt. Mit

```
OUT &FB7F,X
```

und

```
PRINT INP(&FB7F)
```

wird das Datenregister des Controllers beschrieben beziehungsweise gelesen. Das »X« steht dabei für die einzugebenden Werte. Das funktioniert aber nur, wenn der Controller auch dazu bereit ist. Die abgefragten (oder geschriebenen) Werte enthalten zusätzliche Zustandsmeldungen, Befehle für den Controller oder Daten, die zwischen Diskette und Computer übertragen werden sollen.

Diese drei Adressen reichen aus, um die Laufwerke zu steuern. Alle anderen Bauteile des Controllers sind schaltungstechnisches Beiwerk und damit für den Programmierer uninteressant.

Die Controller von Schneider (Amstrad) und Vortex unterscheiden sich zwar in einigen Details, in der Funktion sind sie aber beide gleich. Somit sind alle erwähnten Portadressen identisch. Die folgenden Erklärungen gelten deshalb ohne Einschränkungen für beliebige Kombinationen von CPC-Geräten und Diskettenstationen. Alle beschriebenen Experimente können Sie mit den folgenden drei Programmen vornehmen. Leider erweist sich das eingebaute Basic für solche Untersuchungen als ungeeignet. Deshalb sind alle Listings in Turbo-Pascal geschrieben. Aber auch wer nicht über diesen leistungsfähigen Compiler verfügt, braucht nicht verzweifeln. Auf der Leserservice-Diskette finden Sie die lauffähigen Programme.

»Rundrive« (Listing 1) ist ein umfangreicher Diskettenmonitor. Mit diesem Programm schicken Sie Kommandos

| Befehle des Floppydisc-Controllers µPD765 | | |
|---|---|---|
| **Befehlsname** | **in Kommandophase zu sendende Bytes** | **Resultatphase** |
| READ DATA | <MT MF SK 0 0 1 1 0> <X(5) HD US(2)> C H R N EOT GPL DTL | ST0 ST1 ST2 C H R N |
| READ DELETED DATA | <MT MF SK 0 1 1 0 0> <X(5) HD US(2)> C H R N EOT GPL DTL | ST0 ST1 ST2 C H R N |
| WRITE DATA | <MT MF X 0 0 1 0 1> <X(5) HD US(2)> C H R N EOT GPL DTL | ST0 ST1 ST2 C H R N |
| WRITE DELETED DATA | <MT MF X 0 1 0 0 1> <X(5) HD US(2)> C H R N EOT GPL DTL | ST0 ST1 ST2 C H R N |
| READ A TRACK | <MT MF SK 0 0 0 1 0> <X(5) HD US(2)> C H F N SR GPL DTL | ST0 ST1 ST2 C H R N |
| READ ID | < X MF X 0 1 0 1 0> <X(5) HD US(2)> | ST0 ST1 ST2 C H R N |
| FORMAT A TRACK | < X MF X 0 1 1 0 1> <X(5) HD US(2)> N SC GLF D | ST0 ST1 ST2 U U U U |
| SCAN EQUAL | <MT MF SK 1 0 0 0 1> <X(5) HD US(2)> C H R N EOT GPL STP | ST0 ST1 ST2 C H R N |
| SCAN LOW OR EQUAL | <MT MF SK 1 1 0 0 1> <X(5) HD US(2)> C H R N EOT GPL STP | ST0 ST1 ST2 C H R N |
| SCAN HIGH OR EQUAL | <MT MF SK 1 1 1 0 1> <X(5) HD US(2)> C H R N EOT GPL STP | ST0 ST1 ST2 C H R N |
| SENSE DRIVE STATUS | < 0 0 0 0 0 1 0 0> <X(5) HD US(2)> ST3 | |
| RECALIBRATE | < 0 0 0 0 0 1 1 1> <X(6) US(2)> | |
| SEEK | < 0 0 0 0 1 1 1 1> <X(6) US(2)> NCN | |
| SPECIFY | < 0 0 0 0 0 0 1 1> <SRT(4) HUT(4)> <HLT(7) ND> | |
| SENSE INTERRUPT STATUS | < 0 0 0 0 1 0 0 0> | ST0 PCN |
| INVALID | BAD | DUM |
| INFO: Spitze Klammern »<Bits...>« repräsentieren ein Byte, dessen Wert von Parameterbits abhängt. Ein Wert in runden Klammern hinter der Bezeichnung »(x)« gibt an, aus wievielen Bits sich der Parameter zusammensetzt. | | |

**Tabelle 1. Die Befehle des Controllers µPD765**

an den Controller, wobei alle Auswirkungen am Bildschirm detailliert protokolliert werden. Das Programm übernimmt alle zeitkritischen Routinen. Zum Absturz bringen Sie den Computer nun nicht mehr. Gleichzeitig bleiben aber alle sinnvollen Kommandos wirksam. Sie können also nichts falsch machen und dürfen bedenkenlos experimentieren.

Im Vordergrund steht dabei der Lerneffekt. Sie sehen, was man alles beachten muß, um eine bestimmte einfach erscheinende Operation auszuführen. Mit dem Programm lassen sich aber auch Aufgaben erledigen, denen ein richtiger Diskettenmonitor nicht gewachsen ist. Zum Beispiel können Sie damit Disketten mit 40 Spuren in Laufwerken doppelter Dichte (80 Spuren) lesen (und jede zweite Spur auch umgekehrt) oder bei entsprechendem Laufwerk doppelseitige Disketten mit dem Schneider-Controller. Ein dritter Zweck von Rundrive ist das Ausprobieren von Kommandos, um sie später in anderen Programmen einzusetzen.

Rundrive besteht aus drei Teilen: dem Hauptprogramm aus Listing 1 und den beiden Include-Dateien »Convbyte« (Listing 2) und »Fdc« (Listing 3). Die genaue Funktion wird später erklärt. Zuerst tippen Sie die drei Programme ein und speichern sie unter den Namen »RUNDRIVE.PAS« (Listing 1), »CONVBYTE.INC« (Listing 2) und »FDC.INC« (Listing 3). Die folgenden Experimente können Sie dann sofort in Angriff nehmen.

Das Programm »Readid« (Listing 4) nimmt eine Diskettenanalyse vor und ist damit eine Anwendung dessen, was Sie mit Hilfe von Rundrive lernen.

Ohne Speichererweiterung passen die Programme nicht in einem Stück in den Speicher. Aber keine Angst, Sie brauchen sich für dieses Programm nicht extra eine Speichererweiterung zulegen. Streichen Sie einfach die zwei Zeilen »$i...« (Zeile 9 und 10) aus dem Programm Rundrive und teilen Sie den Rest in zwei ungefähr gleichgroße Stücke auf. Vor der Zeile

```
(* 512 Byte langen...
```

trennen Sie den Code und speichern ihn unter dem Namen »Teil1.INC« und »Teil2.INC«. Auch die Datei Fdc muß an der Stelle

```
(* Diskettenmotoren...
```

geteilt werden. Der Anfang wird mit »FDC1.INC« und das Ende mit »FDC2.INC« gespeichert. Jetzt schreiben Sie ein neues Programm »RUNDRIVE.PAS«, das nur aus einer einzigen Zeile besteht:

```
(*$i FDC1.INC*) (*$i FDC2.INC)
(*$i CONVBYTE.INC*)
(*$i TEIL1.INC*)(*$i TEIL2.INC*)
```

Wählen Sie die Compiler-Option »C« und schon wird das neue Programm auch ohne Speichererweiterung compiliert.

| Symbol | Name | Beschreibung |
|---|---|---|
| BAD | Bad Byte | Einer der 171 nicht als erstes Kommandobyte erlaubten Werte, insbesondere 00 hex |
| C | Cylinder Number | Zylindernummer in der ID des gesuchten Sektors |
| D | Data | Wert, mit dem ein Sektor gefüllt wird |
| DUM | Dummy Byte | Geliefert wird hier immer der Wert 80 hex |
| DTL | Data Length | Wenn N Null ist, gibt DTL die Anzahl der zu übertragenden Bytes an |
| EOT | End of Track | Letzte Sektornummer auf einem Zylinder. Nach dem Lesen/Schreiben eines Sektors dieser Nummer bricht der Controller die Operation ab. |
| F | First Sector | Nummer des ersten Sektors auf der Spur |
| GPL | Gap Length | Zeit, während der der Controller nach dem Lesen/Schreiben eines Sektors Signale von der Diskette nicht beachtet. (Einheit: »Eine Byte-Dauer«) |
| GLF | Gap Length Format | Abstand zwischen zwei Sektoren in Bytes |
| H | Head Address | Kopfadresse in der ID der gewünschten Sektoren |
| HD | Head | gewünschte Kopfnummer des Laufwerks |
| HLT | Head Load Time | Zeit zwischen Aufsetzen des Schreib-/Lesekopfes und Schreib-/Lesebeginn (01=2 ms: ... 7F=254 ms: bei 8MHz Takt (bei 4 MHz 4 ms: ... 508 ms:) |
| HUT | Head Unload Time | Zeit zwischen Ende von Schreiben/Lesen und Abheben des Schreib-/Lesekopfes (01=16 ms: ... 0F=240 ms: bei 8MHz (bei 4 MHz 32 ms: ... 508 ms:) |
| MF | FM- or MFM Mode | 0=Single Density, 1=Double Density |
| MT | Multi Track | 1=nach dem Ende von Seite 0 mit Seite 1 weitermachen, 0=nach dem Ende von Seite 0 Abbruch |
| N | Number | Anzahl der Bytes in einem Sektor 00=128 Byte, 01:=256, 02=512, 03=1024, ... 06:=8192 |
| NCN | New Cylinder Number | Zylinder, auf den der Schreib-/Lesekopf gestellt wird (unabhängig von den Werten in den Ids der Sektoren dieses Zylinders) |
| ND | Non-DMA Mode | 1, wenn Operation nicht mit Hilfe einer DMA abgewickelt wird |
| PCN | Present Cylinder Number | aktuelle Position des Schreib-/Lesekopfes |
| R | Record | ID-Nummer des zu lesenden Sektors |
| SC | Sector | Anzahl der Sektoren pro Zylinder |
| SK | Skip | 1, wenn Sektor mit »Deleted Data Adress Mark« übersprungen wird |
| SR | Sectors Read | Anzahl der zu lesenden Sektoren |
| SRT | Step Rate Time | Zeit, die der Schreib-/Lesekopf zum Spurwechseln braucht (0F=1 ms:, 0E=2 ms:, ..., 00=16 ms: bei 8 MHz Takt) (bei 4 MHz 2 ms: ... 32 ms:) |
| ST0 | Status 0 | Inhalt des Statusregisters 0 |
| ST1 | Status 1 | Inhalt des Statusregisters 1 |
| ST2 | Status 2 | Inhalt des Statusregisters 2 |
| ST3 | Status 3 | Inhalt des Statusregisters 3 |
| STP | Step | Sektorversatz beim Durchsuchen mehrerer Sektoren (R<alt> + STP → R<nächst>) |
| U | Undefined | Diese Werte haben keine Bedeutung |
| US | Unit Select | Welches Laufwerk wurde gewählt (0,...,3)? |
| X | | beliebiger Wert, in der Regel steht hier aber 0 |

**Tabelle 2. Byte-Namen in Klarschrift**

## Was kann der μPD765?

Der μPD765 kann bis zu vier Laufwerke mit den Nummern 0 bis 3 steuern. Diese Nummern haben jedoch nichts mit den Kennbuchstaben für die Laufwerke zu tun. Ein Beispiel dafür ist das dritte Laufwerk bei dem Vortex-Controller. Die eingebauten Laufwerke benutzen die Geräteadressen 0 und 1. Ein Schneider-3-Zoll-Laufwerk liegt normalerweise ebenfalls auf einer dieser Geräteadressen. Der bei Vortex erhältliche Adapterstecker macht nichts anderes, als die Anschlußleitungen so raffiniert zu vertauschen, daß beim Aufruf der Adresse 3 die Nummer 1 oder 0 resultiert. Das Kommando »S2« schaltet nicht etwa durch dubiose





**Status-Register 0**

| Bit | Name | Bedeutung |
|---|---|---|
| D7-D6 | Interrupt Code | 00: Kommando korrekt ausgeführt, normales Ende<br>01: gestartetes Kommando nicht erfolgreich beendet<br>10: Ungültiges Kommando angefordert. Nicht gestartet<br>11: Abbruch, weil Ready-Signal vom Laufwerk geändert |
| D5 | Seek End | 1, nachdem SEEK-Kommando vervollständigt |
| D4 | Equipment Check | 1, wenn Laufwerk Fehler meldet oder bei RECALIBRATE nach 77 Schritten Spur 0 nicht gefunden wurde |
| D3 | Not Ready | 1, wenn ein Schreib-/Lesekommando angefordert wurde und Laufwerk noch nicht Ready meldete oder nach Versuch, Kopf 2 eines Einzelkopflaufwerkes anzusprechen |
| D2 | Head Address | Nummer des betroffenen Kopfes |
| D1-D0 | Unit Select | Nummer des betroffenen Laufwerkes |

**Status-Register 1:**

| Bit | Name | Bedeutung |
|---|---|---|
| D7 | End of Cylinder | 1 nach Versuch, Sektor hinter Spurende anzusprechen |
| D6 | | nicht verwendet, immer 0 |
| D5 | Data Error | CRC-Lesefehler im ID- oder Datenfeld eines Sektors |
| D4 | Over Run | Controller vom Programm während Datenübertragung nicht schnell genug bedient |
| D3 | | nicht verwendet, immer 0 |
| D2 | No Data | Nach READ DATA, WRITE DELETED DATA oder SCAN:<br>1, wenn angeforderter Sektor nicht gefunden wurde<br>Nach READ ID:<br>1, wenn ein ID-Feld nicht fehlerfrei gelesen<br>Nach READ A TRACK:<br>1, wenn Sektor nicht fortlaufend numeriert ist |
| D1 | Not Writable | 1, wenn Laufwerk während WRITE DATA, WRITE DELETED DATA oder FORMAT A TRACK Schreibschutz meldet. |
| D0 | Missing Address Mark | 1, wenn das Indexloch zweimal entdeckt wurde, bevor ein ID-Feld oder ein Datenfeld gefunden wurde. Im letzten Fall wird zugleich Bit 0 von ST2 gesetzt. |

**Status-Register 2:**

| Bit | Name | Bedeutung |
|---|---|---|
| D7 | | nicht verwendet, immer 0 |
| D6 | Control Mark | 1, wenn während READ DATA oder SCAN ein Sektor mit einer Deleted Data Address Mark entdeckt wurde |
| D5 | Data Error in Data Field | 1, wenn im Datenfeld eines Sektors ein CRC-Lesefehler auftrat |
| D4 | Wrong Zylinder | 1, wenn die angeforderte Spurnummer mit der auf der Diskette verzeichneten nicht übereinstimmte (ähnlich Bit 2 von Register 1) |
| D3 | Scan Equal Hit | 1, wenn während eines SCAN-Kommandos die Bedingung »gleich« erfüllt war |
| D2 | Scan Not Satisfied | 1, wenn während eines SCAN-Kommandos kein Sektor gefunden wurde, der die Bedingung erfüllte |
| D1 | Bad Cylinder | 1, wenn der Sektor auf der Diskette die Spurnummer FF enthält und eine andere Spurnummer angefordert wurde (ähnlich Bit 1 von Status-Register 1) |
| D0 | Missing Address Mark in Data Field | 1, wenn beim Lesen eines Sektors zwar eine ID Address Mark, aber keine Data Address Mark oder Deleted Data Address Mark gefunden wurde |

**Status-Register 3:**

| Bit | Name | Bedeutung |
|---|---|---|
| D7 | Fault | Zeigt den Zustand des Fehlersignals des angesprochenen Laufwerks an |
| D6 | Write Protected | Zustand des Schreibschutz-Signals vom Laufwerk |
| D5 | Ready | Ready-Signal des Laufwerks |
| D4 | Track 0 | Spur-0-Signal vom Laufwerk |
| D3 | Two Side | Signal, ob Laufwerk doppelseitig |
| D2 | Head Address | Welcher Kopf des Laufwerks wurde angesprochen? |
| D1-D0 | Unit Select | Welches Laufwerk wurde angesprochen? |

**Tabelle 3. Die Bedeutung der Statusregister**

Tricks das eingebaute Laufwerk ab, sondern setzt nur ein Flag (Merker).

Beim nächsten Zugriff auf das Laufwerk B wird durch die Abfrage dieses Merkers festgestellt, ob nun die Schreib-/Leseroutinen für das Laufwerk an der Geräteadresse 3 oder für das Laufwerk an der Adresse 1 bestimmt sind. Mit entsprechender Software können Sie somit auch ein viertes Laufwerk anschließen. Beim Schneider-Controller ist ein ähnlicher Trick leider nicht ohne Hardware-Basteleien möglich, da die Entwickler die notwendigen Anschlußleitungen eingespart haben.

Der $\mu$PD765 unterstützt zwei verschiedene Aufzeichnungsverfahren, die sich in der Codierung der Bits unterscheiden. Das erste trägt den Namen »IBM 3740 single density format« (oder »FM«), das andere »IBM-System 34 double density format« oder auch »MFM«. Da die Unterschiede beim Programmieren nicht in Erscheinung treten, sondern nur, wenn Sie den Laufwerken mit einem Oszilloskop zuleibe rücken, gehen wir hier nicht näher darauf ein.

In der Praxis kommt meist das neuere MFM-Verfahren zur Anwendung, weil man damit durch einige ausgeklügelte Tricks bei der Codierung in einem bestimmten Abschnitt der Diskette bei gleicher Signalfrequenz doppelt soviele Daten wie beim FM-Verfahren unterbringt. Daneben erfolgt die Datenübertragung doppelt so schnell.

Die beiden Aufzeichnungsverfahren stehen jeweils in zwei Varianten zur Verfügung: mit den Frequenzen 250 KHz und 500 KHz. Die Auswahl erfolgt, indem ein Anschluß eines ICs im Controller entweder mit Masse oder mit 5 Volt verbunden wird. Dadurch wird der $\mu$PD765 entweder mit 4 MHz oder mit 8 MHz getaktet. Leider liegt dieser Anschluß bei den Schneider-Computern ständig auf Masse, so daß nur die Taktfrequenz von 4 MHz und damit eine Signalfrequenz von 250 KHz möglich ist. Die höhere Aufzeichnungsgeschwindigkeit wird allerdings auch nur bei 8 Zoll und bei einigen äußerst exotischen $5^1/_4$-Zoll-Laufwerken eingesetzt. International hat sich die niedrigere Geschwindigkeit durchgesetzt, so daß es nur wenige Computer gibt, deren Disketten von Laufwerken am Schneider PC nicht gelesen werden können. Vom Computer aus gesehen, verhalten sich übrigens Laufwerke in verschiedenen Formaten (also 3-, $3^1/_2$- oder $5^1/_4$-Zoll) vollkommen identisch.

Zur Datenübertragung zwischen Computer und Controller unterstützt der $\mu$PD765 drei Verfahren: Bei der ersten Methode generiert der Controller, sobald ein Byte übertragen werden soll (egal in welcher Richtung), einen

Interrupt. Bei den Geräten von Schneider ist aber die Interruptleitung des Controllers nicht mit dem Computer verbunden. Diese Methode darf also nicht eingesetzt werden.

Der zweite Weg dient zur Zusammenarbeit mit einem DMA-Controller (DMA = Direkt Memory Access). Beim Übertragen von langen Dateien mit vielen Bytes vertrödelt die CPU die meiste Zeit damit, festzustellen, daß tatsächlich nichts weiter gemacht werden soll, als ein Byte zu übertragen, und mit der Kontrolle, ob die Übertragung zu Ende ist. Ein DMA-Baustein leistet nun außer dem Übertragen von Bytes überhaupt nichts anderes. Die oben erwähnten Entscheidungen der CPU braucht und kann er gar nicht treffen. Die Folge ist, daß die Datenübertragung mit DMA bei einem Z80 10- oder 20mal schneller erfolgt als ohne DMA. Leider gibt es bis heute für die Schneider-Computer noch kein Zusatzgerät mit eingebauter DMA. Damit fällt auch diese Steuerung weg.

Bei den Schneider-Computern bleibt also nur noch die dritte, die »Polling-Methode«. Hierbei meldet der µPD765 im Statusregister, daß ein Byte übertragen werden soll. Die CPU testet dann in einer Schleife ständig, ob Daten zum Übertrag vorliegen.

Der µPD765 überprüft nicht, ob bei den einzelnen Operationen die Diskettenmotoren eingeschaltet sind oder nicht. Darum müssen Sie sich selbst kümmern. Klar ist, daß die Motoren bei allen Operationen, die Daten lesen oder schreiben, eingeschaltet sein müssen. Aber auch beim Aufsuchen einer bestimmten Spur muß diese Bedingung erfüllt sein. Bei allen anderen Operationen dürfen die Motoren allerdings ausgeschaltet bleiben. Nach dem Einschalten der Motoren müssen Sie (beziehungsweise Ihr Programm) den Motoren etwas Zeit lassen, um richtig in Schwung zu kommen. Sonst erhalten Sie ständig scheinbar unerklärliche Fehlermeldungen. Im Normalfall ist bei den Schneider-Computern eine Wartezeit von einer Sekunde voreingestellt. Sie können aber auch mit anderen Zeiten experimentieren. Beim Ausschalten erweist es sich als günstig, den Motor interruptgesteuert noch einige Sekunden nachlaufen zu lassen. Bei einem kurzfristigen nächsten Einschalten spart das dann die Hochlaufzeit.

Der µPD765 kennt 15 Kommandos, die in Tabelle 1 aufgelistet sind. Alle Kommandos sind nach demselben Schema aufgebaut. Jedes besteht aus einer Kommandophase, dem Ausführungsbereich und dem Ergebnis. In der Kommandophase werden zwischen einem und neun Byte vom Computer an den Controller übertragen. Sie geben an, welches Kommando ausgeführt werden soll. Das erste Byte sagt dabei, welches Kommando angesprochen ist (zum Beispiel »Lies Sektoren«). Daraus ergibt sich die Zahl der weiter zu übertragenden Bytes, die nötig sind, um das Kommando genauer zu beschreiben (zum Beispiel welchen Sektor).

In der nächsten Phase wird die Operation ausgeführt. Es werden entweder Daten von der CPU an den µPD765 (beim Schreiben), vom Controller an die CPU (beim Lesen) oder auch gar nichts übertragen. Sobald die Operation abgeschlossen ist, folgt die Ergebnisphase. Dabei werden einige Statusbytes vom

```
(*******************************************************************)
(*   Experimentierhilfe zum Direktzugriff auf Disketten-Controller µPD765   *)
(*                     - ein 'Diskettenmonitor' -                           *)
(*               hardwareunabhaengige Teile des Programms                   *)
(*                          (Vers. 16.07.1986)                              *)
(*******************************************************************)

(* Einbinden der Module, die das Programm benoetigt                         *)

{$i fdc.inc}  (* hardwareabhaengige Teile der Laufwerksteuerung      *)
{$iconvbyte.inc} (* verschiedene Konvertierungen von Bytes zu Strings   *)

const vers: string[8] = '16.07.86'; (* aktuelle Programmversion        *)
      maxtoken = 20;      (* maximale Argumentanzahl in der Eingabezeile    *)
      maxbuffer = 10239;  (* Datenpufferlaenge fuer Datenuebertragung       *)

type line = string[80];                    (* Eingabezeilen                 *)
     token = array[0..maxtoken] of integer; (* dekodierte Eingaben          *)
     comtab = array[0..8] of byte;          (* Befehlsbytes FDC-Kommando    *)
     exectab = array[0..maxbuffer] of byte; (* Datenpuffer fuer Laufwerk    *)
     restab = array[0..6] of byte;       (* Zustandsmeldung des Controllers *)

var command: comtab;       (* Raum fuer Kommandobytes fuer Controllerbefehl  *)
    execute: exectab;      (* Raum fuer von/an zu uebertragende Bytes        *)
    result: restab;        (* Raum fuer Zustandsmeldung des Controllers      *)
    werte: token;          (* dekodierte Eingaben                            *)
    eingabe: line;         (* Eingabezeilen                                  *)
    befehl: char;          (* Welcher Befehl wird gewuenscht?                *)
    start,                 (* Ab welchem Byte wird Puffer gefuellt?          *)
    fehler: integer;       (* Fehlerposition in der Eingabezeile             *)
    busy: byte;            (* Welche Laufwerke sind gerade aktiv?            *)

(* Eingabezeile untersuchen, Kommando und Hexzahlen heraussuchen *)
procedure scanline(var s:line; var c:char; var w: token; var fehler: integer);
var laenge, anfang, ende, i:integer;
begin
  c := upcase(s[1]);                       (* Befehlscode zurueckgeben *)
  fehler := 0; i := 0;                     (* kein Fehler, kein Schluesselwort *)
  laenge := length(s);                            (* Laenge der Eingabezeile  *)
  ende := 2;                                      (* erste Zahl suchen        *)
  while s[ende] = ' ' do ende := succ(ende);      (* Space nicht beachten     *)
  while (ende <= laenge) and (i <= maxtoken) and (fehler = 0) do
    begin
      anfang := ende;                              (* Ende der Zahl suchen    *)
      ende := anfang - 1 + pos(' ',copy(s, anfang, laenge));
      if ende = anfang - 1                 (* wenn nicht gefunden, dann bis Ende *)
        then ende := laenge + 1;
      i := succ(i);          (* Zahl hexadezimal betrachten und Wert ablegen *)
      val('$' + copy(s,anfang,ende-anfang), w[i], fehler);
      while s[ende] = ' ' do ende := succ(ende);    (* Space nicht beachten *)
    end;
  if fehler <> 0 then begin i := pred(i); fehler := anfang end;
  w[0] := i                                  (* Anzahl der Befehlsargumente *)
end;

(* Disketten-Operation ausfuehren mit Anzeige der Ergebnisse *)
procedure operate(var werte: token; var command: comtab; var execute: exectab;
  var result: restab; var busy: byte);
var i: integer;
begin
  if werte[0] > 9 then werte[0] := 9;      (* hoechstens 9 Argumente erlaubt *)
  for i:= 1 to werte[0] do command[i - 1] := werte[i]; (* in Commandtabelle *)
  fillchar(result, 7, 0);                      (* Zustands-Tabelle loeschen *)
  fdcinterrupt(0);                                    (* Interrupts sperren  *)
  busy := fdccall(command[0],execute[0],result[0]);  (* Disketten-Operation *)
  fdcinterrupt(1);                                    (* Interrupts erlauben *)
  write('Command: ');                             (* Kommandobytes anzeigen *)
  for i:= 0 to 8 do write(convhex(command[i]), ' '); writeln; (* hexadezimal*)
  write('Result:  ');                          (* Controllerzustand ausgeben *)
```

**Listing 1. »Rundrive« – eine Experimentierhilfe**

```
  for i:= 0 to 6 do write(convhex(result[i]), '       '); writeln;   (* Hex *)
  write('         ');
  for i:= 0 to 6 do write(convbin(result[i]), ' '); writeln;      (* binaer *)
  writeln('Busy:     ', copy(convbin(busy), 5, 4));   (* Drives beschaeftigt *)
end;

(* 512 Byte langen Datenpufferausschnitt anzeigen, ab Versatzadresse *)
procedure showbuffer(var werte: token; var execute: exectab);
var i, start, zeichen: integer;
begin
  if werte[0] = 0 then start := 0           (* ohne Argumente Standardpuffer *)
    else start := werte[1] shl 9;   (* Puffernummer mal 512 (=Pufferlaenge) *)
  write('Execute: ');
  for i := start to start + 511 do        (* 512 Byte eines Puffers ausgeben *)
    begin
      zeichen := execute[i] and 127;             (* Steuerzeichen ausblenden *)
      if (zeichen < 32) or (zeichen = 127)
        then write('.') else write(chr(zeichen));
      if (i and 63) = 63 then                       (*  64 Byte in einer Zeile *)
        begin writeln; write('         ') end
    end
end;

(* Diskettenmotoren ein- oder ausschalten *)
procedure motor(var werte: token);
begin
  if werte[0] = 0 then fdcmotor(0)             (* ohne Argumente ausschalten *)
    else fdcmotor(werte[1])                    (* sonst wahlweise            *)
end;

(* gewaehlten Datenpuffer loeschen *)
procedure clearbuffer(var werte: token; var execute: exectab; start: integer);
begin
  if werte[0] = 0 then werte[1] := 0;     (* ohne Argument mit Null loeschen *)
  fillchar(execute[start], 512, werte[1])    (* Loeschen ab aktuellem Puffer *)
end;

(* gewaehlten Datenpuffer mit bestimmten Bytes beschreiben *)
procedure setbuffer(var werte: token; var execute: exectab; start: integer);
var i:integer;
begin                                (* Argumente in aktuellen Puffer kopieren *)
  for i := 1 to werte[0] do execute[start + i - 1] := werte[i]
end;

(* gewaehlten Datenpuffer mit Text beschreiben *)
procedure writebuffer(var eingabe:line; var execute: exectab; start :integer);
begin  (* Eingabezeile ab zweitem Zeichen in aktuellen Datenpuffer kopieren *)
  move(eingabe[2], execute[start], length(eingabe) - 1)
end;

(* 512 Byte-Datenpuffer in anderen Datenpuffer kopieren *)
procedure copybuffer(var werte: token; var execute: exectab);
begin                        (* nur wenn mindestens Quellpuffer angegeben *)
  if werte[0] <> 0 then if werte[0] = 1
    then move(execute[werte[1] shl 9], execute, 512)    (* in ersten Puffer *)
    else move(execute[werte[1] shl 9], execute[werte[2] shl 9], 512)
end;          (* aus Puffernummer im zweiten Argument Zieladresse berechnen *)

(* zu aendernden Datenpuffer waehlen *)
procedure selectbuffer(var werte: token; var start: integer);
begin            (* wenn Argument, dann Puffernummer mal 512(=Pufferlaenge) *)
  if werte[0] = 0 then start := 0 else start := werte[1] shl 9
end

(* Alle erlaubten Eingaben auflisten *);
procedure help;
begin
  writeln;
  writeln('Experimentierhilfe fuer Disketten-Controller µPD765 und aehnliche');
  writeln('  (Vers. ', vers, ', angepasst fuer ', computertyp, ')');
  writeln;
  writeln('erlaubte Eingaben: (alle Zahlen hexadezimal)');
  writeln('  Add              -> 512 Byte-Datenpuffer auswaehlen');
  writeln('  Cd1 d2 d3 .. d9 -> Kommando an Disketten-Controller uebergeben');
  writeln('  Ddd              -> 512 Byte-Datenpuffer Nr. d1 anzeigen');
  writeln('  E                -> Programmende');
  writeln('  Fdd              -> gewaehlten Datenpuffer mit Wert d1 fuellen');
  writeln('  H                -> Hilfe');
  writeln('  Kd1 d2           -> Datenpuffer d1 in Datenpuffer d2 kopieren');
  writeln('  Mdd              -> 0 = Laufwerksmotoren abschalten, sonst ein');
  writeln('  Sd1 d2 d3 ..     -> Bytes in gewaehlten Datenpuffer schreiben');
```

**Listing 1. »Rundrive« – eine Experimentierhilfe (Fortsetzung)**

Controller an die CPU übertragen. Der Inhalt dieser Bytes gibt detailliert darüber Auskunft, ob die Operation erfolgreich war und wenn nicht, welche Fehler auftraten. Bei einigen Operationen gibt es allerdings keine Ergebnisphase.

Der Controller nimmt es mit den in der Kommando- und der Ergebnisphase zu übertragenden Bytes sehr genau. Sobald ein einziges Byte abgeschickt wurde, müssen auch alle anderen übertragen werden. Wenn es sich das Programm nach der Hälfte der Daten »anders überlegt«, so wartet der Controller bis zum Ausschalten auf die fehlenden Bytes. In der Ergebnisphase müssen ebenso alle Bytes abgeholt werden, bevor ein neues Kommando angefordert wird.

Das Statusregister des µPD765 darf zwischen Kommando- und Ergebnisbytes erst nach einer Wartezeit von 0,5 µsek abgefragt werden. Ganz weglassen dürfen Sie diese Abfrage aber auch nicht. Bei Kommandos ohne Ergebnisphase dauert es nach dem letzten Datensatz einige Zeit, bis das Kommando tatsächlich beendet ist. Um zu vermeiden, daß das Programm noch ein weiteres Byte überträgt, das dann vom Controller als nächstes Kommando interpretiert wird, muß also eine Warteschleife eingefügt werden. Trotz Mindestwartezeit muß man aber auch das Maximum beachten. Beim Übergang von der Kommandophase zur Ausführungsphase ist die maximale Ausführungszeit wieder recht knapp bemessen. Unkritisch in bezug auf die Anzahl der zu übertragenden Bytes ist allein die Ausführungsphase. Dafür muß sich hier das Programm sputen, damit die mit einer Geschwindigkeit von einer Viertelmillion Bit pro Sekunde übertragenen Bytes rechtzeitig verarbeitet werden. Denn sonst gehen einige Informationen verloren. Zudem kann sich das Programm auch hier aufhängen, beispielsweise wenn es darauf wartet, ein Byte zum Controller senden zu können, während dieser gerade Bytes abliefert.

Im Hauptstatusregister meldet der µPD765 ständig, in welchem Zustand er sich gerade befindet. Bit 7 ist gesetzt, wenn der Controller auf ein Byte wartet oder ein Byte zum Abholen bereitliegt. Bit 6 gibt an, in welche Richtung die durch Bit 7 gemeldete Datenübetragung erfolgen soll. Ist Bit 6 Null, so wartet der Controller auf ein Byte vom Prozessor. Andersherum liegt ein Byte zum Abholen bereit. Bit 5 hat während der Befehlsausführung den Wert 1, sonst den Wert 0. Bit 4 hat den Wert 1, während ein Kommando in Arbeit ist (also die ganze Zeit zwischen dem ersten Kommandobyte und dem letzten

Ergebnisbyte). Bit 0, 1, 2 oder 3 haben den Wert 1, wenn der Schreib-/Lesekopf des Laufwerks auf der Geräteadresse 0, 1, 2 oder 3 dabei ist, eine neue Spur aufzusuchen, die Anforderung des Spurwechsels aber schon beendet ist.

Reicht es Ihnen nun schon von den ganzen Bits und davon, wann welches warum den Wert 1 oder 0 annimmt? Das dachten wir uns auch und schrieben deshalb ein kleines Unterprogramm, das diesen ganzen »Kleinkram« erledigt. Dem Programm müssen nur drei Tabellen übergeben werden. Wie die Tabellen zu behandeln sind, entscheidet es selbst. Es verläßt sich dabei ganz auf die Statusbits, die der µPD765 liefert.

Solange der Controller Kommandobytes anfordert, liefert das Programm diese aus der Kommandotabelle (auch auf die Gefahr hin, daß ein paar unsinnige dabei sind). Danach überträgt es die Daten zwischen Ausführungstabelle und Controller, wobei der Controller die Richtung angibt (auch wenn der Programmierer eine andere erwartete), bis diese Phase zu Ende ist (auch wenn einige nicht benötigte Bytes gelesen oder bedeutungslose geschrieben werden). Zum Schluß werden solange Bytes in die Ergebnistabelle übertragen, bis der Controller das Kommandoende meldet (auch wenn die Bytes wegen eines unsinnigen Kommandos nicht erwünscht sind).

Während eines Befehls ist der Controller also Herr über den Computer. Für diese Freiheit revanchiert er sich damit, daß sich der Computer niemals aufhängt und nach dem Ende des Unterprogramms der Controller sofort für ein neues Kommando bereitsteht. Die Zahl der übertragenen Bytes hält sich dabei allerdings in Grenzen. Als Ergebnis liefert das Unterprogramm den Zustand der Bits 0 bis 3 des Hauptstatusregisters, also welche Schreib-/Leseköpfe gerade eine neue Spur aufsuchen. Die Bits 4 bis 7 haben nur innerhalb des Unterprogramms eine Bedeutung und werden deshalb mit dem Wert 0 zurückgegeben. Nur ein Fehler kann somit theoretisch auftreten: Wenn der Controller am Anfang des Unterprogramms nicht bereit ist, ein Kommando zu übernehmen, so gibt das Unterprogramm den Wert FF hex zurück. Das kommt aber in der Regel nicht vor.

Aus Zeitgründen ist es notwendig, die Bedienungsroutine für den Controller in Assembler zu schreiben. Für nicht so zeitkritische Experimente verwendet man aber bequemer eine höhere Programmiersprache. Damit haben auch Nicht-Assemblerkundige die Chance, eigene Experimente vorzunehmen.

In der Datei »Fdc« ist deshalb das

```
    writeln('  Tttttttttttt..  -> Text in gewaehlten Datenpuffer schreiben');
    writeln;
    writeln('    ',convhex((maxbuffer+1)shr 9),'h 512 B-Datenpuffer verfuegbar');
end;

(* Hauptprogramm *)
begin
  help;
  start := 0; fillchar(command, 9, 0);   (* Kommando loeschen und Puffer 0 *)
  repeat
    writeln; write('===> '); readln(eingabe);
    scanline(eingabe, befehl, werte, fehler);
    case befehl of
      'A': selectbuffer(werte, start);
      'C': operate(werte, command, execute, result, busy);
      'D': showbuffer(werte, execute);
      'E': writeln('----- Ende -----');
      'F': clearbuffer(werte, execute, start);
      'H': help;
      'K': copybuffer(werte, execute);
      'M': motor(werte);
      'S': setbuffer(werte, execute, start);
      'T': writebuffer(eingabe, execute, start);
      else writeln('***** falsche Eingabe *****')
    end
  until befehl = 'E';
end.
```

**Listing 1. »Rundrive« – eine Experimentierhilfe (Schluß)**

```
(* Byte in binaere Schreibweise umwandeln *)
type binstring=string[8];
function convbin(z:byte):binstring;
var erg:binstring;begin
inline($3a/z/$21/erg/$4f/$06/$08/$70/$23/$af/$cb/$11/$ce/$30/$77/$10/$f7);
convbin:=erg end;

(* Byte in hexadezimale Schreibweise umwandeln *)
type hexstring=string[2];
function convhex(z:byte):hexstring;
var erg:hexstring;begin
inline($21/erg/$3a/z/$36/$02/$4f/$1f/$1f/$1f/$1f/$e6/$0f/$c6/$90/$27/$ce/$40/
$27/$23/$77/$79/$e6/$0f/$c6/$90/$27/$ce/$40/$27/$23/$77);
convhex:=erg end;
```

**Listing 2. »Convbyte« wird Wandlung genannt**

```
(**************************************************************************)
(*       Funktionen zum Direktzugriff auf den Disketten-Controller µPD765  *)
(*                       fuer Z80-Computersysteme                          *)
(* hier: Einbindung der Maschinensprache passend fuer Turbo-Pascal, CP/M-80 *)
(*              (c) by Isar-Amper-Soft (Vers. 06.08.1986)                  *)
(**************************************************************************)

(* notwendige Ein/Ausgabe-Portadressen zum Steuern der Laufwerke:         *)
(*                                                                         *)
(*  FDCPOR: Statusregister Disketten-Controller                            *)
(*          (Datenregister hat automatisch Adresse (FDCPOR + 1)            *)
(*  MOTPOR: Portadresse zum Steuern der Diskettenmotoren                   *)
(*  MOTON:  Datenbyte zum Motoreinschalten (ueber Port MOTPOR)             *)
(*  MOTOFF: Datenbyte zum Motorausschalten (ueber Port MOTPOR)             *)

const FDCPOR: integer = $fb7e;   (* Adresswerte an eigene Hardware anpassen *)
      MOTPOR: integer = $fa7e;   (* hier: Adressen fuer Schneider CPC,      *)
      MOTON: byte = $01;         (* Schneider- oder Vortex-Controller       *)
      MOTOFF: byte = $00;

const computertyp: string(.13.) = 'Schneider CPC';    (* Name des Computers *)

(* Laufwerk-Operation ausfuehren                                           *)
(* comman, execut und result muessen Elemente eines Feldes vom Typ          *)
(* "array(.0..max.) of byte" sein. Die zu uebertragenden Bytes werden in    *)
(* dem Feld aufeinanderfolgend hinter dem angegebenen Feldelement abgelegt. *)

function fdccall(var comman, execut, result: byte):byte;
var busy: byte;
```

**Listing 3. »Fdc« – Direktangriff auf den Controller**

```
begin
Inline(           (* ;****** Vorspann zum Einsatz unter Turbo-Pascal          *)
$ED/$4B/FDCPOR/ (*          ld bc,(fdcpor)         ;1. Arg (Controllport)   *)
$ED/$5B/COMMAN/ (*          ld de,(comman)         ;2. Arg (Command-Tab)    *)
$2A/RESULT/     (*          ld hl,(result)         ;4. Arg (Result-Tab)     *)
$E5/            (*          push hl                ;liegt auf Stapel        *)
$2A/EXECUT/     (*          ld hl,(execut)         ;3. Arg (Execute-Tab)    *)
                (* ;****** Ausfuehren der Controller-Operation               *)
                (* ;(Vorbereitung der Operation)                              *)
$ED/$78/        (* wait00: in a,(c)                                          *)
$87/            (*          add a,a                ;Uebertragung moeglich?  *)
$30/$11/        (*          jr nc,wait3c           ;nein -> warten          *)
$E6/$E1/        (*          and OE1 hex            ;jetzt Kommando fertig?  *)
$3E/$FF/        (*          ld a,OFF hex           ;Flag fuer Fehler        *)
$20/$62/        (*          jr nz,fdcerr           ;darf nicht vorkommen    *)
                (* ;(Kommandophase: Ausgabeschleife)                          *)
$1A/            (* wait1c: ld a,(de)               ;Byte aus Tabelle holen  *)
$13/            (*          inc de                                           *)
$0C/            (*          inc c                  ;Datenregister FDC       *)
$ED/$79/        (*          out (c),a              ;Byte senden             *)
$0D/            (*          dec c                  ;Statusregister FDC      *)
$3E/$05/        (*          ld a,5                 ;kurze Pause             *)
$3D/            (* wait2c: dec a                   ;(Zeit zur Verarbeitung) *)
$20/$FD/        (*          jr nz,wait2c                                     *)
$ED/$78/        (* wait3c: in a,(c)                                          *)
$87/            (*          add a,a                ;Uebertragung moeglich?  *)
$30/$FB/        (*          jr nc,wait3c           ;nein -> warten          *)
$87/            (*          add a,a                ;Datenrichtung lesen?    *)
$38/$21/        (*          jr c,wait3e            ;ja -> Lese oder Result *)
$87/            (*          add a,a                ;Execute-Phase?          *)
$38/$0D/        (*          jr c,wait1e            ;ja -> Schreibschleife   *)
$3E/$0A/        (*          ld a,10                ;kurze Pause             *)
$3D/            (* wait4c: dec a                   ;(Zeit zum Kommando-     *)
$20/$FD/        (*          jr nz,wait4c           ;abschluss benoetigt)    *)
$ED/$78/        (*          in a,(c)                                         *)
$E6/$10/        (*          and 10 hex             ;Kommando abgeschlossen? *)
$20/$DF/        (*          jr nz,wait1c           ;nein -> Kommandobyte    *)
$18/$3B/        (*          jr fdcend              ;Kommandoende            *)
                (* ;(Ausfuehrungsphase: Schreibschleife)                      *)
$7E/            (* wait1e: ld a,(hl)               ;zu sendendes Byte       *)
$0C/            (*          inc c                  ;Datenregister FDC       *)
$ED/$79/        (*          out (c),a              ;senden                  *)
$0D/            (*          dec c                  ;Statusregister FDC      *)
$23/            (*          inc hl                 ;naechstes Byte          *)
$ED/$78/        (* wait2e: in a,(c)                ;Status holen            *)
$F2/*-3/        (*          jp p,wait2e            ;Warten bis bereit       *)
$E6/$20/        (*          and 20 hex             ;Ende der Uebertragung   *)
$20/$F1/        (*          jr nz,wait1e           ;nein -> weiter senden   *)
$18/$12/        (*          jr wait1r              ;Result-Phase            *)
                (* ;(Ausfuehrungsphase: Test, ob Leseschleife noetig)         *)
$87/            (* wait3e: add a,a                 ;Ausfuehrungsphase?      *)
$30/$0F/        (*          jr nc,wait1r           ;nein -> Resultschleife *)
                (* ;(Ausfuehrungsphase: Leseschleife)                         *)
$0C/            (* wait4e: inc c                   ;Datenregister FDC       *)
$ED/$78/        (*          in a,(c)               ;Byte empfangen          *)
$0D/            (*          dec c                  ;Statusregister FDC      *)
$77/            (*          ld (hl),a                                        *)
$23/            (*          inc hl                 ;naechstes Byte          *)
$ED/$78/        (* wait5e: in a,(c)                ;Status holen            *)
$F2/*-3/        (*          jp p,wait5e            ;Warten bis bereit       *)
$E6/$20/        (*          and 20 hex             ;Ende der Uebertragung?  *)
$20/$F1/        (*          jr nz,wait4e           ;nein -> weiteres lesen *)
                (* ;(Result-Phase: Vorbereitung)                              *)
$E3/            (* wait1r: ex (sp),hl              ;Tabelle Result-Phase    *)
$ED/$78/        (* wait2r: in a,(c)                ;Status lesen            *)
$87/            (*          add a,a                                          *)
$30/$FB/        (*          jr nc,wait2r           ;Warten bis bereit       *)
$E6/$20/        (*          and 20h                ;Operation beendet?      *)
$28/$0D/        (*          jr z,wait4r            ;ja -> Ende  Result      *)
$0C/            (*          inc c                  ;Datenregister FDC       *)
$ED/$78/        (*          in a,(c)               ;Result-Byte holen       *)
$0D/            (*          dec c                  ;Statusregister FDC      *)
$77/            (*          ld (hl),a              ;Byte in Tabelle         *)
$23/            (*          inc hl                                           *)
$3E/$04/        (*          ld a,4                 ;kurze Pause             *)
$3D/            (* wait3r: dec a                   ;(Zeit zur Verarbeitung) *)
$20/$FD/        (*          jr nz,wait3r                                     *)
$18/$EA/        (*          jr wait2r              ;Result-Schleife         *)
$E3/            (* wait4r: ex (sp),hl              ;Ausfuehrungs-Tab zurueck*)
                (* ;(Rueckgabe vorbereiten)                                   *)
$ED/$78/        (* fdcend: in a,(c)                ;Hauptstatusregister     *)
```

**Listing 3. »Fdc« – Direktangriff auf den Controller (Fortsetzung)**

Assemblerprogramm so abgelegt, daß das Resultat eine reine Turbo-Pascal-Funktion ist. Programmieren Sie gerne in Assembler, so ziehen Sie die Kommentarzeilen aus dem Programm und verwenden diese als Assemblerquelldatei weiter. Leider gilt die hier benutzte Art der Parameterübergabe nur für Turbo-Pascal. Wenn Sie einen anderen Compiler benutzen, kommen Sie um einige kleine Anpassungen nicht herum.

Als Parameter dürfen der Funktion keine Felder übergeben werden. Die erlaubten Bytes müssen aus Elementen von Tabellen des Typs »array [.0..max.] of byte« stammen. Wenn Daten von der Diskette zum Beispiel ab dem 1000. Byte der Ausführungstabelle abgelegt werden sollen, geben Sie beim Aufruf an:

```
busy := callfdc(command(.0.),
execute(.1000.), result(.0.))
```

Wenn die Daten ab dem ersten Byte abgelegt werden, geben Sie

```
busy := callfdc(...,
execute(.0.),...)
```

an. Sind Sie sicher, daß ein Kommando keine Ausführungs- oder Ergebnisphase besitzt, können Sie anstelle der Tabelle eine beliebige Variable vom Typ Byte übergeben:

```
busy := callfdc(command(.0.),
dummy, dummy)
```

Die Funktion »callfdc« funktioniert auch bei eingeschalteten Interrupts. Während der Ausführungsphase kommt es dann aber zu Zeitproblemen. Ein geschriebener Sektor sieht so aus, als ob das Laufwerk »stottert«. Andererseits kann es sinnvoll sein, auch während unkritischer Kommandos die Interrupts abzuschalten, beispielsweise um keine wichtige Stelle der Diskette zu übersehen. Deshalb enthält die Datei »FDC.INC« auch eine Prozedur zur Behandlung der Interrupts. Beachten Sie aber, daß alle Betriebssystemaufrufe (Bildschirmausgabe, Tastaturabfrage, normale BDOS-Diskettenoperationen) die Interrupts wieder einschalten. Bei abgeschalteten Interrupts läßt sich übrigens der Computer nicht einmal mehr mit <SHIFT+CTRL+ESC> stoppen.

Wahrscheinlich werden Sie diese Datei mit

```
(*$i FDC.INC*)
```

sehr oft in eigene Programme einbinden. Die Zeit zum Compilieren verkürzt sich in diesem Fall wesentlich, wenn Sie eine »Kurzform« der Datei erstellen, in der alle Kommentare fehlen. Ferner sollte jede Zeile bis zum Ende genutzt werden. Diese Kurzform können Sie mit »^KR« auch direkt in Ihr Programm einfügen.

»Rundrive« erlaubt eine sehr komfortable Programmierung des Disket-

ten-Controllers. Wenn Sie als ersten Buchstaben einer Zeile ein »c« eingeben, werden die nachfolgenden hexadezimalen Ziffern als in der Kommandophase einer Diskettenoperation zu sendende Bytes verstanden. Die entsprechende Routine wird sofort ausgeführt und die Ergebnisbytes sowie der Laufwerkszustand nach der Operation dezimal und hexadezimal ausgegeben. Das Kommando dürfen Sie sehr weit abkürzen. Wenn sich von einem Kommando zum nächsten etwa nur das erste Byte ändert, brauchen Sie bei der Eingabe auch nur dieses angeben. Wollen Sie das beim nächsten Mal das gleiche Kommando noch einmal starten, tippen Sie nur ein »c« ein. Erfolgten dazwischen keinerlei andere Eingaben, genügt es sogar, nur <ENTER> zu drücken.

Andere Kommandos unterstützen diese Funktion ebenfalls. Sie dürfen Texte und hexadezimale Werte in einen von 20 Datenpuffern schreiben, den Datenpuffer löschen, kopieren oder anzeigen. Selbstverständlich lassen sich die Diskettenmotoren zu jedem beliebigen Zeitpunkt ein- und ausschalten. Sogar eine »Hilfe«-Funktion ist vorhanden. Es fehlt also nichts, was Sie zum Steuern Ihrer Laufwerke benötigen.

Die Programmdatei Rundrive ist vollständig hardwareunabhängig. Die drei verwendeten hardwareabhängigen Funktionen »callfdc«, »motorfdc« und »interruptfdc« beinhaltet die Datei »Fdc«. Damit der Compiler diese auch kennt, müssen sie mit Hilfe von

```
(*$i FDC.INC*)
```

in das Programm integriert werden. So reicht es aus, die Funktionen nur ein einziges Mal zu schreiben und schon stehen sie in jedem Programm bereit. »Fdc« enthält dabei auch eine Konstante, die den Namen des Computertyps angibt.

Weiterhin benötigt wird die Datei »Convbyte«. Diese enthält zwei Funktionen: zum Umwandeln eines Bytes in eine hexadezimale und eine dezimale Zeichenkette. Genauso wie »Fdc« kann sie universell eingesetzt werden.

Die 15 Controllerbefehle stehen in der Tabelle 1. Jede Abkürzung in der Tabelle repräsentiert ein zu übertragendes Bit oder Byte. Die Bedeutung der Abkürzungen erklärt Tabelle 2.

Manche Kommandobytes setzen sich aus verschiedenen zusammen, wobei jedes Bit eine andere Einzelheit des Kommandos regelt. In diesem Fall stehen alle Bits, die für ein Kommando notwendig sind, in spitzen Klammern. Wenn zur Regelung eines bestimmten Details des Kommandos mehrere Bits nötig sind, steht hinter der Bezeichnung (in runden Klammern eingeschlossen) die Anzahl der Bits. Beim

```
$E6/$0F/           (*        and 0F hex              ;Drivestatusbits      *)
                   (* fdcerr:                                              *)
                   (* ;****** Nachspann fuer Verwendung unter Turbo-Pascal *)
$C1/               (*        pop bc                  ;Result-Tab vom Stapel *)
$32/BUSY)          (*        ld (busy),a             ;an Pascal liefern    *)
                   (* ;      (bei Turbo-Pascal kein ret)                   *)
;fdccall:=busy
end;

(* Diskettenmotoren ein- oder ausschalten (0 = aus, sonst ein)            *)

procedure fdcmotor(flgmot: byte);
begin
Inline(            (* ;****** Vorspann zur Verwendung unter Turbo-Pascal   *)
$ED/$4B/MOTPOR/ (*           ld bc,(motpor)          ;1. Arg (I/O-Adresse)  *)
$3A/MOTON/         (*        ld a,(moton)            ;2. Arg (Einschaltwert) *)
$5F/               (*        ld e,a                                        *)
$3A/MOTOFF/        (*        ld a,(motoff)           ;3. Arg (Ausschaltwert) *)
$57/               (*        ld d,a                                        *)
$3A/FLGMOT/        (*        ld a,(flgmot)           ;4. Arg (Ein oder Aus) *)
$A7/               (*        and a                                         *)
                   (* ;****** Ausfuehren der Operation                     *)
$7A/               (*        ld a,d                                        *)
$28/$01/           (*        jr z,motx               ;Z, dann ausschalten  *)
$7B/               (*        ld a,e                  ;nein, dann einschalten *)
$ED/$79)           (* motx:  out (c),a               ;Motor ein/ausschalten *)
                   (* ;****** Nachspann zur Verwendung unter Turbo-Pascal  *)
                   (* ;      (bei Turbo-Pascal kein ret)                   *)
end;

(* Interrupts sperren oder erlauben (0 = sperren, sonst freigeben)        *)

procedure fdcinterrupt(flgint: byte);
begin
Inline(            (* ;****** Vorspann zur Verwendung unter Turbo-Pascal   *)
$3A/FLGINT/        (*        ld a,(flgint)        ;1. Arg (Ein oder Aus)   *)
$A7/               (*        and a                                         *)
                   (* ;****** Ausfuehren der Interruptwahl                 *)
$28/$03/           (*        jr z,disint             ;Z, dann sperreb      *)
$FB/               (*        ei                      ;Interrupts erlauben  *)
$18/$01/           (*        jr endint                                     *)
$F3)               (* disint: di                     ;Interrupts sperren   *)
                   (* endint:                                              *)
                   (* ;****** Nachspann zur Verwendung unter Turbo-Pascal  *)
                   (* ;      (bei Turbo-Pascal kein ret)                   *)
end;
```

**Listing 3. »Fdc« – Direktangriff auf den Controller (Schluß)**

```
program readid;
(************************************************************************)
(*                       Diskettenanalyse                               *)
(*          - findet alle formatierten Sektoren einer Diskette -        *)
(*                        (Vers. 2.08.86)                               *)
(************************************************************************)

{$i fdc.inc}       (* Funktionen zum Direktzugriff auf Disketten-Controller *)
{$i convbyte.inc}  (* Funktionen zum Umwandeln von Bytes in Zeichenketten   *)

const maxid = 31;
type resultfeld = array[0..maxid, 0..6] of byte; msgstr = string[80];
var drive, track, side: byte; twoside: char; endofdisk: boolean;

procedure specify;
const command: array[0..2] of byte = (3, 0, 3);
var speed: integer; dummy: byte;
begin
  repeat
    write('Spurwechselzeit in msec? (1 bis 32) '); readln(speed);
  until (speed >= 1) and (speed <= 32);
  command[1] := 257 - (speed + (speed and 1)) shl 3;
  dummy := fdccall(command[0], dummy, dummy)
end;

procedure abbruch(message: msgstr);
begin writeln('**** Fehler: ', message, ' ****');fdcmotor(0);specify;halt end;

procedure home(drive: byte);
```

**Listing 4. »Readid« findet alle Sektoren**

```
const calibrate: array[0..1] of byte = (7, 0); sense: byte = 8;
var dummy: byte; result: array[0..1] of byte;
procedure primitivehome;
  begin dummy := fdccall(calibrate[0], dummy, dummy);
        repeat until fdccall(sense, dummy, result[0]) = 0; end;
begin
  calibrate[1] := drive and 3;
  repeat until (fdccall(sense, dummy, result[0]) = 0) and (result[0] = 128);
  primitivehome; if (result[0] and 192) <> 0 then primitivehome; delay(32);
  if(result[0] and 192) <> 0 then abbruch('Justierung nicht erfolgreich')
end;

procedure seektrack(drive, zylinder: byte);
const seek: array[0..2] of byte = (15, 0, 0); sense: byte = 8;
var dummy: byte; result: array[0..1] of byte;
begin
  repeat until (fdccall(sense, dummy, result[0]) = 0) and (result[0] = 128);
  seek[1]:=drive and 3;seek[2]:=zylinder;dummy:
=fdccall(seek[0],dummy,dummy);
  repeat until fdccall(sense, dummy, result[0]) = 0; delay(32);
  if(result[0] and 192) <> 0 then abbruch('Zylinder nicht gefunden')
end;

procedure readids(drive, side, dens: byte; var anzahl, start: byte;
  var idfeld: resultfeld);
var readid: array[0..1] of byte; result: array[0..6] of byte; dummy: byte;
begin
  fdcinterrupt(0);
  readid[0]:=(dens and 1)shl 6+10;readid[1]:=(drive and 3)+(side and 1)shl 2;
  dummy := fdccall(readid[0], dummy, result[0]); anzahl := 0; start := 0;
  if (result[0] and 192) = 0 then
    repeat
      dummy := fdccall(readid[0], dummy, idfeld[anzahl,0]);
      if idfeld[anzahl, 5]<idfeld[start,5]then start:=anzahl;anzahl:=anzahl+1
    until idfeld[anzahl - 1, 5] = result[5];
  fdcinterrupt(1);
end;

procedure auswahl(var drive: byte; var twoside: char);
var continue: char;
begin
  write('Welches Laufwerk analysieren? (0 bis 3) ');
  repeat read(kbd,continue) until (continue >= '0') and (continue <= '3');
  writeln(continue); drive := ord(continue) - ord('0');
  write('Doppelseitig analysieren ? (J/N) ');
  read(kbd, twoside); twoside := upcase(twoside); writeln(twoside);
  write('Bitte Diskette einlegen und dann eine Taste druecken ');
  repeat until keypressed; writeln;
  writeln; write('---Position--    ');
  writeln('-----------------Werte in den Sektor-IDs-----------------')
end;

function spurinhalt(drive, track, side: byte): boolean;
var continue, density: char; idfeld: resultfeld; max, st, i: byte;
begin
  spurinhalt := false;
  density := 'D'; readids(drive, side, 1, max, st, idfeld);
  if max=0 then begin density:='S';readids(drive,side,0,max,st,idfeld)end;
  if max = 0
    then begin write('nicht formatiert. Weitersuchen? (J/N) ');
               read(kbd,continue); write(upcase(continue));
               if upcase(continue)<>'J' then spurinhalt := true; end
    else begin
           write(density, 'D Track ', convhex(idfeld[0,3]), ' Head ');
           write(convhex(idfeld[0,4]), ' Size ', idfeld[0,6], ' Sek');
            for i:=st to st+max-1 do write(' ',convhex(idfeld[i mod max,5]))
         end;
  writeln
end;

begin
  writeln('Diskettenanalyse'); writeln('---------------'); writeln;
  specify; auswahl(drive,twoside); fdcmotor(1); delay(1000); home(drive);
  track := 0; side := 0;
  repeat
    write('Cyl ', convhex(track), ' Side ', side, ' -> ');
    if side = 0 then seektrack(drive, track);
    endofdisk := spurinhalt(drive, track, side);
    if twoside='J'then side:=(side+1)and 1;if side=0 then track:=track+1
  until endofdisk;
  home(drive); fdcmotor(0); specify
end.
```

**Listing 4. »Readid« findet alle Sektoren (Schluß)**

Kommando »Write Data« heißt das erste Kommandobyte zum Beispiel »<MT MF 0 0 0 1 0 1>«. Bit 7 (MT, das höchstwertige Bit) regelt, ob nach dem Beschreiben der ersten Seite mit der zweiten Seite fortgefahren werden soll. Ist dies nicht der Fall, muß dieses Bit auf 0 gesetzt werden. Bit 6 (MF) bestimmt, welches Aufzeichnungsverfahren angewandt wird. In diesem Fall soll es das MFM-Verfahren sein, weshalb dieses Bit auf 1 gesetzt wird. Insgesamt ergibt das erste Kommandobyte also den Wert 01000101 bin oder 45 hex. Das zweite Byte heißt »<X(5) HD US(2)>«. »X(5)« bedeutet, daß die fünf höchstwertigen Bits beliebige Werte annehmen dürfen. »HD« muß den Wert Null haben, wenn auf Seite Null geschrieben werden soll und »US(2)« gibt an, daß in den zwei niederwertigsten Bits die Geräteadresse des anzusprechenden Laufwerks steht. Wenn auf Seite Null des Laufwerks 1 geschrieben werden soll, muß das zweite Kommandobyte also den Wert XXXXX001 bin haben, also zum Beispiel 01 hex oder F9 hex.

Die meisten Kommandos fordern die Angabe von Zylindernummer, Kopfnummer, Sektornummer und Sektorlänge (C, H, R, N). Als Zylindernummer wird nicht die Position des Schreib-/Lesekopfes angegeben, sondern die Nummer, die sich in den Sektormarkierungen auf der angesprochenen Diskette befindet. Die beiden Nummern können damit zwar übereinstimmen, eine Notwendigkeit ist es aber nicht. Das ist zum Beispiel bei Disketten mit 40 Spuren, die in 80-Spur-Geräten liegen, der Fall oder bei doppelseitigen Formaten, bei denen in den Sektormarkierungen statt der physikalischen Zylindernummer die logische Spurnummer steht. Sinngemäß gilt das gleiche für die anderen drei Byte der Sektormarkierungen (oder Sektor-IDs). Die Ergebnisphase liefert ebenfalls die Bytes C, H, R und N.

ST0, ST1, ST2 und ST3 geben die Inhalte der Statusregister des Controllers an. Aus diesen können Sie ablesen, ob die letzte Operation erfolgreich war oder welcher Fehler auftrat. Die Bedeutungen der einzelnen Bits ersehen Sie aus Tabelle 3. Die Bits 0 bis 2 in den Statusregistern 0 und 3 geben an, auf welches Laufwerk sich die Inhalte der Bits 3 bis 7 beziehen. Fehler wie »Daten nicht gefunden« treten auf, wenn nach zweimaligem Passieren des Indexloches (eine volle Diskettenumdrehung) auf der Diskette die angeforderte Sektormarkierung nicht gefunden wurde.

(Helmut Tischer/hg)

# Disketten – eine runde Sache

**Forscher ans Werk! Experimentieren Sie mit dem Disketten-Controller. Lesen Sie Daten aus Bereichen, die sonst nur dem Profi zugänglich sind!**

Nach der Theorie des vorhergehenden Beitrags geht es jetzt los mit dem Experimentieren. Arbeiten Sie aber nur mit einer Diskette, deren Daten Sie nicht mehr benötigen. Es passiert öfter als man denkt, daß mit der Floppy »nichts mehr geht«.

Als erstes teilen Sie dem Controller alle Leistungsdaten des Laufwerks mit. Die »Head Load Time« gibt an, wieviel Zeit vergehen muß zwischen dem Hand-Load-Signal (dem Befehl an das Laufwerk, den Schreib-/Lesekopf auf die Diskette zu senken) und dem ordnungsgemäßen Lesen beziehungsweise Schreiben. Die »Head Unload Time« gibt an, nach welcher Zeit der Schreib-/Lesekopf nach dem letzten Zugriff von der Diskette abgehoben werden darf. Wenn während dieser Zeit ein neuer Zugriff erfolgt, wird die »Head Load Time« eingespart, da der Kopf noch auf der Diskette ist.

Die Schneider-Controller beachten dieses Signal nicht. Der Schreib-/Lesekopf bleibt auf der Diskette, solange der Motor eingeschaltet ist. Das Specify-Kommando (zuständig für das Einstellen der Laufwerksdaten) wählt deshalb die kürzest erlaubten Zeiten, nämlich 4 beziehungsweise 32 Millisekunden. Das Bit »Non-DMA Mode« gibt an, ob die Operationen mit oder ohne einem DMA (Baustein zur direkten Datenübertragung) ausgeführt werden. Bei Schneider hat dieses Bit immer den Wert 1.

## Schwung mit DMA

Aus der »Step Rate Time« ersehen Sie, wie lange der Schreib-/Lesekopf des Laufwerks von einer Spur zur nächsten braucht. Der Controller richtet sich danach und schickt die Schrittimpulse nicht schneller zum Laufwerk, als dieses sie ausführen kann. Zwischen 2 und 32 (in Schritten von jeweils 2) Millisekunden läßt sich dieser Wert einstellen. Bemerkenswerterweise wird hier jedoch »rückwärts« gezählt, der Wert 0F entspricht also 2 Millisekunden und der Wert 00 hex 32 Millisekunden. Die Vortex-Laufwerke arbeiten mit einer Schrittzeit von 3 (Einstellen lassen sich aber nur 4 Millisekunden), die Schneider-Stationen hingegen mit 12 Millisekunden. Manche 3-Zoll-Laufwerke vertragen aber auch Schrittzeiten bis herunter zu 6 Millisekunden. Etwas Experimentieren schadet also nichts. Die Standardeinstellung erhalten Sie mit dem Programm Rundrive (siehe Seite 132) mit »C3 E1 3« beziehungsweise »C3 A1 3« (Option C (Kommando an Disketten-Controller übergeben)).

## Aufsuchen einer Spur

Mit dem Controller-Kommando »Recalibrate« wird das angegebene Laufwerk auf den Zylinder 0 justiert. Dabei fahren die Schreib-/Leseköpfe solange in Richtung Zylinder 0, bis je nach Gerätetyp eine Lichtschranke oder ein mechanischer Anschlag erreicht wird. Wenn der Zylinder die Spur 0 nach 77 Schrittimpulsen nicht gefunden hat, erfolgt eine Fehlermeldung. Bei Laufwerken mit 80 Zylindern (Spuren) muß dieses Kommando damit immer zweimal hintereinander ausgeführt werden.

Das Kommando »Seek« positioniert die Schreib-/Leseköpfe des angegebenen Laufwerks über dem gewünschten Zylinder. Diese Position ist von den Sektormarkierungen der eingelegten Diskette unabhängig.

Nachdem eines dieser beiden Kommandos gestartet wurde, meldet das entsprechende Laufwerk im Hauptstatusregister solange »Busy«, bis das Kommando »Sense Interrupt Status« aufgerufen wird. Erst danach sind auf diesem Laufwerk wieder Schreib-/Lesebefehle zugelassen.

Während der Wartezeit, bis der gesuchte Zylinder gefunden wurde, gibt der Controller die Kontrolle an den Computer zurück. Diese Zeit kann also genutzt werden, um andere Kommandos zu starten. So können zum Beispiel auf allen vier Laufwerken (so viele kann der µPD765 maximal kontrollieren) gleichzeitig »Seek«-Operationen ausgeführt werden. Nach dem Erreichen des richtigen Zylinders wird vor dem ersten Zugriff eine kleine Wartepause eingelegt, damit der Schreib-/Lesekopf vollständig zum Stillstand kommt. Das normale Betriebssystem setzt hier 32 Millisekunden ein.

Geben Sie also im Programm Rundrive »M1« ein. Damit starten Sie die Diskettenmotoren. Wenn Sie jetzt »C7 0« eingeben, hören Sie, wie der Schreib-/Lesekopf vom Laufwerk A in die Ausgangspositon fährt. »Cf 0 10« stellt ihn auf den Zylinder 16.

Zylinder und Spur sind zwei nah verwandte Begriffe. Es gibt aber einen Unterschied. Die Zylindernummer gibt die absolute Position auf der Diskette an, unabhängig davon, was an dieser Stelle steht – also zum Beispiel »10 Schrittimpulse von der Ausgangsposition«. Mit »Spur« ist dagegen die logische Spur gemeint, die auch unter CP/M bekannt ist. Wenn Sie jetzt eine Diskette mit 40 Spuren in ein Laufwerk mit 80 Spuren einlegen, befindet sich jeweils nur auf einem geradzahligen Zylinder eine logische Spur (wobei die Spurnummer dann meist genau den halben Wert der Zylindernummer besitzt). Ferner enthält bei einer doppelseitigen Diskette ein Zylinder zwei logische Spuren: eine auf der Oberseite der Diskette und eine auf der Unterseite. Auch hier nimmt deshalb die auf der Diskette markierte Spurnummer einen anderen Wert an als der Zylinder.

## Laufwerkszustand abfragen

Wenn auf einem Laufwerk ein »Recalibrate«, oder ein »Seek«-Kommando beendet wurde oder die Ready-Leitung eines Laufwerks den Zustand änderte, generiert der Controller-Baustein µPD765 einen Interrupt. Danach kann man mit dem Kommando »Sense Interrupt Status« feststellen, welches Laufwerk den Interrupt verursacht hat und aus welchem Grund. Nach diesem Kommando wird der Interrupt gelöscht.

Leider leitet der µPD765 bei den Schneider-Computern die Interrupts nicht an die CPU weiter. Es bleibt also nichts anderes übrig, als von Zeit zu Zeit den Interruptstatus probeweise abzufragen. Glücklicherweise bemerkt der Controller die Abfrage des Interruptstatus auch wenn gar kein Interrupt vorlag. In diesem Fall steht in dem Register ST0 der Wert 80 hex, das Register PCN wird überhaupt nicht abgefragt.

Beachten Sie, daß man den Status für jeden Interrupt extra abfragen muß. Wenn beispielsweise auf allen vier Lauf-

werken ein »Seek«-Kommando beendet wurde und sich das Ready-Signal vom Laufwerk Null änderte, muß man fünfmal hintereinander das »Sense«-Kommando aufrufen. Jedesmal ergibt sich dann der Zustand eines anderen Laufwerks. Erst bei der sechsten Abfrage erhält man die Meldung 80 hex für Interrupt gelöscht.

Das Ready-Signal eines Laufwerks ändert sich, wenn beim Einschalten des Motors eine bestimmte Drehzahl erreicht wird und zuvor eine Diskette eingelegt war. Einige Laufwerkstypen melden mit diesem Signal auch, wenn eine Diskette gewechselt wurde. Ein solches Laufwerk kann nach Wunsch bei unerlaubtem Diskettenwechsel eine Warnung auf dem Bildschirm ausgeben.

Mit »Sense Drive Status« lassen sich einige weitere Informationen über ein wählbares Laufwerk abfragen.

## Formatieren einer Spur

Bevor eine Diskette einsatzbereit ist, muß sie formatiert werden. Dabei werden an allen Stellen, an denen später Sektoren stehen sollen, Markierungen angebracht. Das gewährleistet, daß ein bestimmter Sektor auch bei mehrmaligem Überschreiben immer dieselbe Stelle der Diskette beibehält und sich nicht mit einem anderen Sektor überlappt. In der Kommadophase beim Formatieren müssen drei wichtige Parameter bestimmt werden: die Länge jedes Sektors, die Anzahl der Sektoren und der Abstand zwischen den Sektoren in Bytes. Das nächste Byte gibt das Zeichen an, mit dem die formatierten Sektoren beschrieben werden sollen. Dieser Wert ist im Prinzip beliebig. Üblicherweise wählt man aber E5 hex. Bei vielen Computertypen invertiert die Schreib-/Leseroutine alle Bytes, bevor sie sie an den Anwender oder an den Controller weitergibt. Unter Pascal macht dies

```
datenbyte := datenbyte xor 255
```

rückgängig. Für solche Formate ist das Füllbyte mit dem Wert 1A hex angebracht.

In der Ausführungsphase müssen Sie dem µPD765 für jeden Sektor der Spur vier Byte übergeben. Das erste der vier Byte gibt die Spurnummer an, das zweite die Kopfnummer, das dritte die Sektornummer und das vierte die Sektorlänge. Diese Bytes fließen in die Sektormarkierung auf der Diskette ein. Die ersten drei Byte können Sie dabei beliebig wählen. Allerdings empfiehlt es sich, auf einer Spur die einmal gewählte Spur- und Kopfnummer beizubehalten. Sonst kommen Sie beim Bearbeiten dieser Sektoren leicht in Schwierigkeiten. Es vereinfacht spätere Programme, wenn die formatierte Spurnummer sowohl zur Zylindernummer als auch zur logischen Spurnummer in einer möglichst unkomplizierten Beziehung steht. Bei einseitig genutzten Disketten verwendet man deshalb meistens die Kopfnummer 00 hex, für die Rückseite einer doppelseitigen Diskette 01 hex. Die Sektorennummern sind beliebig. Am besten wählen Sie sie aus einem zusammenhängenden Wertebereich. Die Reihenfolge der Sektoren sollten Sie so einrichten, daß sich zwischen zwei logisch aufeinanderfolgenden Sektoren immer ein weiterer Sektor befindet. Während der unwichtige Sektor unter dem Schreib-/Lesekopf vorbeistreicht, hat dann das Anwendungsprogramm genügend Zeit, den gelesenen Sektor zu verarbeiten. Wenn die nächsten Informationen gelesen werden sollen, steht der passende Sektor fast augenblicklich zur Verfügung. Ohne diesen Trick ist der gesuchte Sektor gerade vorbei, und das Programm muß eine volle Diskettenumdrehung warten, bis der betreffende Sektor das nächstemal gelesen wird. Bei Formaten mit neun Sektoren hat sich die folgende Reihenfolge bewährt:

```
01 06 02 07 03 08 04 09 05
```

Diese Reihenfolge bezeichnet man als »Sektorversatz 2«. Beachten Sie aber, daß der Nummernabstand zwischen zwei physikalisch benachbarten Sektoren dabei nicht etwa ebenfalls 2, sondern 5 beträgt.

Der Wert im Byte »Sektorgröße« sollte bei Standardformaten dem in der Kommandophase angegebenen Wert entsprechen – also 0 für eine Sektorlänge von 128 Byte, 1 für 256 Byte, 2 für 512 Byte und so weiter. Sie dürfen aber auch jederzeit einem einzelnen Sektor eine andere Größenangabe mitgeben. Mit einem Schreibbefehl auf diesen Sektor wird dann die neue Anzahl Daten geschrieben und eventuell der nächstfolgende Sektor überschrieben. So kann man verschieden lange Sektoren auf einer einzigen Spur erzeugen.

## Der Sektorabstand

Ein schwieriges Problem blieb bisher ausgeklammert: Zu einer gegebenen Länge den richtigen Abstand zwischen den Sektoren und die maximale Sektor-

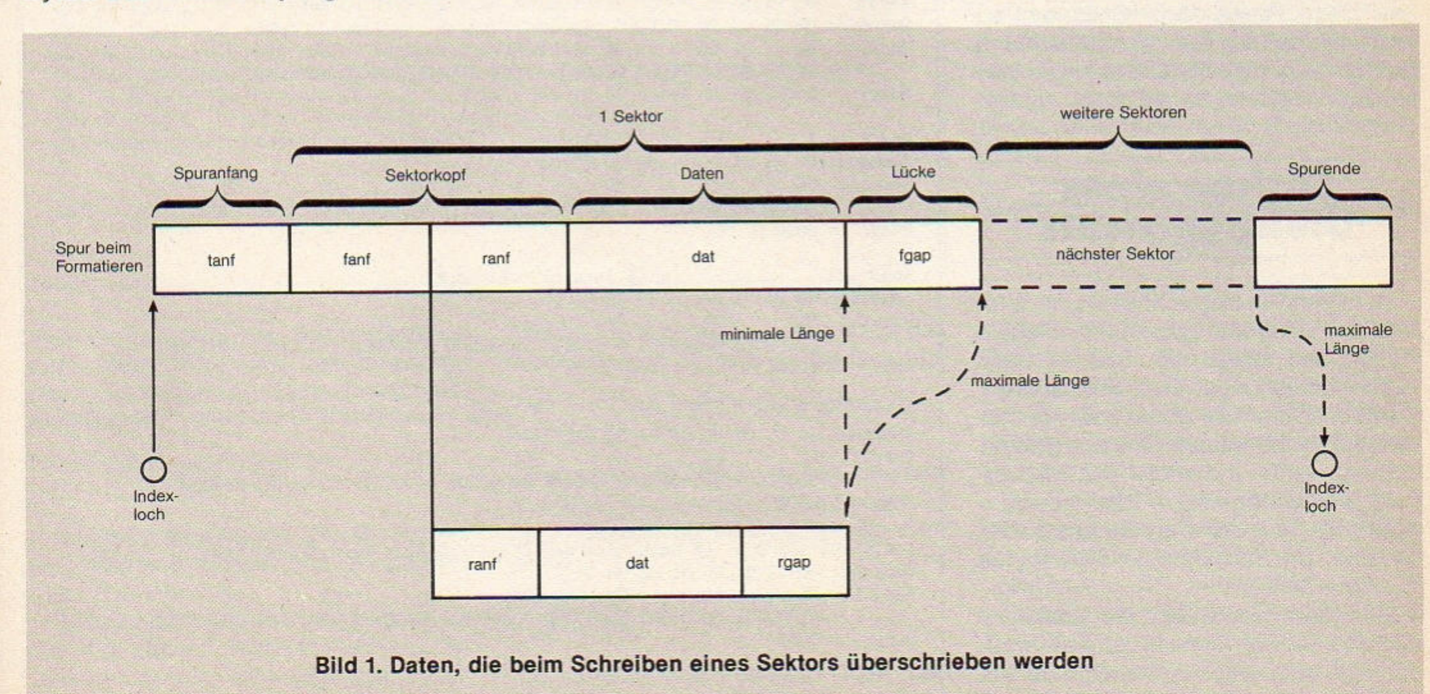



Bild 1. Daten, die beim Schreiben eines Sektors überschrieben werden

zahl zu finden. Zu diesem Thema gibt es leider keine offiziellen Informationen. Einen Anhaltspunkt bieten aber die Beispiele im Datenblatt zum µPD765. Diese finden Sie in der Tabelle.

Diskettenlaufwerke, die für eine Signalfrequenz von 250 kHz geeignet sind, laufen mit fünf Umdrehungen pro Sekunde. Deshalb passen bei dem FM-Verfahren genau 3125 Byte auf eine Spur; bei dem MFM-Verfahren sogar 6250 Byte. Allerdings reserviert der µPD765 am Anfang einer Spur 146 Byte zur eigenen Verwendung. Jeder Sektor enthält außer den Daten noch einen Kopf, der unabhängig von der Zahl Daten 62 Byte lang ist. Im Sektorkopf stehen zum Beispiel die Markierungen, die beim Formatieren angegeben wurden, interne Markierungen, Prüfsummenbytes und Signale zur Synchronisation der Lesevorgänge mit der Datenaufzeichnung. Die 62 Byte teilen sich auf in 22 Byte, die nur einmal beim Formatieren geschrieben und 40 Byte, die bei jedem Überschreiben erneuert werden. Beim Formatieren ist der Abstand der Sektoren so groß zu wählen, daß trotz Toleranzen bei der Umdrehungsgeschwindigkeit der Disketten beim Beschreiben eines Sektors der nächstfolgende Sektor nicht berührt wird. Trotzdem müssen natürlich alle Sektoren auf die Diskette passen.

Beim Schreiben, beziehungsweise Lesen, muß ebenfalls ein Abstand angegeben werden. Während der Zeit, in der sich das Ende eines zu lang geratenen – und überschriebenen – Sektors unter dem Lesekopf befindet, dürfen diese gelieferten Daten nicht beachtet werden. Diese könnten sonst fälschlicherweise als Anfang des neuen Sektors interpretiert werden. Andererseits darf diese Pause aber auch nicht so lang sein, daß der Anfang des nächsten Sektors verpaßt wird. Die optimale Verteilung der Daten zeigt Bild 1.

## Suchen eines Gleichgewichts

Im folgenden bezeichnet »t« die Toleranz der Umdrehungsgeschwindigkeit, »rgap« und »fgap« die Lücken beim Lesen beziehungsweise Formatieren, »ranf« und »fanf« die immer neu oder nur einmal geschriebenen Teile des Sektorkopfes, »tanf« und »tlen« der Spuranfang beziehungsweise die gesamte Spurlänge, »n« und »dat« die Anzahl der Sektoren pro Spur oder der Datenbytes in einem Sektor.

Damit beim Schreiben oder Lesen der nächste Sektor niemals verpaßt wird, dürfen die neu geschriebenen Teile des Sektors zusammen mit der Lücke nie-

```
Experimentierhilfe fuer Disketten-Controller µPD765 und aehnliche
  (Vers. 16.07.86, angepasst fuer Schneider CPC)

erlaubte Eingabemoeglichkeiten: (alle Zahlen hexadezimal)
Add              -> 512 Byte-Datenpuffer auswaehlen
Cd1 d2 d3..d9-> Kommando an Disketten-Controller uebergeben
Ddd              -> 512 Byte-Datenpuffer Nr. d1 anzeigen
E                -> Programmende
Fdd              -> gewaehlten Datenpuffer mit Wert d1 fuellen
H                -> Hilfe
Kd1 d2           -> Datenpuffer d1 in Datenpuffer d2 kopieren
Mdd              -> 0 = Floppymotoren abschalten, sonst einschalten
Sd1 d2 d3 ..     -> Bytes in gewaehlten Datenpuffer schreiben
Ttttttttttt..    -> Text in gewaehlten Datenpuffer schreiben

    14h 512 B-Datenpuffer verfuegbar

===> c3 e1 3            (* Laufwerksdaten angeben *)
Command: 03 E1 03 00 00 00 00 00 00
Result:  00       00       00       00       00       00       00
         00000000 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0000

===> m1                 (* Motor einschalten *)

===> c7 1               (* Laufwerk 1 (entspricht B) justieren *)
Command: 07 01 03 00 00 00 00 00 00
Result:  00       00       00       00       00       00       00
         00000000 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0010

===>                    (* Noch einmal *)
***** falsche Eingabe *****

===> c                  (* der Kommentar war schuld ! *)
Command: 07 01 00 00 00 00 00 00 00
Result:  00       00       00       00       00       00       00
         00000000 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0010

===> c8                 (* Interruptstatus abfragen *)
Command: 08 01 03 00 00 00 00 00 00
Result:  C0       3C       00       00       00       00       00
         11000000 00111100 00000000 00000000 00000000 00000000 00000000
Busy:    0010

===> c                  (* solange bis alle Interrupts gemeldet *)
Command: 08 01 03 00 00 00 00 00 00
Result:  21       00       00       00       00       00       00
         00100001 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0000

===>
Command: 08 01 03 00 00 00 00 00 00
Result:  80       00       00       00       00       00       00
         10000000 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0000

===> cf 1 1             (* Zylinder 1, Laufwerk 1 aufsuchen *)
Command: 0F 01 01 00 00 00 00 00 00
Result:  00       00       00       00       00       00       00
         00000000 00000000 00000000 00000000 00000000 00000000 00000000
Busy:    0010

===> c8                 (* Interruptstatus *)
Command: 08 01 01 00 00 00 00 00 00
Result:  21       01       00       00       00       00       00
         00100001 00000001 00000000 00000000 00000000 00000000 00000000
Busy:    0000

===> c46 1 1 0 0 2 9 2a ff        (* Spur 2, entspricht Zyl 1 Seite 1 lesen *)
Command: 46 01 01 00 00 02 09 2A FF
Result:  41       04       00       01       00       00       02
         01000001 00000100 00000000 00000001 00000000 00000000 00000010
Busy:    0000

===> c46 1 1 0 1  (* Huch, das war falsch. Gleich korrigieren! *)
Command: 46 01 01 00 01 02 09 2A FF
Result:  41       80       00       01       00       09       02
         01000001 10000000 00000000 00000001 00000000 00001001 00000010
Busy:    0000
```

```
===> m          (* Motor aus *)

===> d         (* Spannung --- wurde wohl das Inhaltsverzeichnis gelesen? *)
Execute: .GRADEMO4PAS....................HELLO   BAS..................
         .$OSC    SYS...0................COPY    COM..................
         .CALL    COM....................FILECOPYCOM..................
         .CASCOPY COM....................PARCOPY COM..................
         .SYSCOPY COM....................SYSGEN  COM..................
         .INSTALL COM...D................VDOS    COM..................
         .S2      COM....................SO      COM..................
         .LOAD    COM....................ASM     COM...@..............

===> d1     (* Ja! Und weil's so schoen war, noch einen Sektor ansehen *)
Execute: .XSUB    COM....................SUBMIT  COM..................
         .DUMP    COM....................ED      COM...4..............
         .STAT    COM...)................DDT     COM...&..............
         .PIP     COM...:................CPM61   COM...' 'a...........
         .LETTER  COM....!...............DIR128  COM....".............
         .PATCH   COM....#$%&'...........SPTEST  COM....(.............
         .FILCOP62COM....)...............RAMDISK COM....*.............
         .SPOOL   COM....+...............COPY62  COM....,.............

===> e (* eigentlich wurden neun Sektoren gelesen. Fuers erste reicht's aber!*)
----- Ende -----
```

**Bild 2. Eine Sitzung mit »Rundrive«**

```
Diskettenanalyse
----------------

Spurwechselzeit in ms? (1 bis 32) 4
Welches Laufwerk analysieren? (0 bis 3) 1
Doppelseitig analysieren? (J/N) J
Bitte Diskette einlegen und dann eine Taste druecken

---Position--    -----------------Werte in den Sektor-IDs-----------------
Cyl 00 Side 0 -> DD Track 00 Head 00 Size 2 Sek 00 08 03 06 01 09 04 07 02 05
Cyl 00 Side 1 -> DD Track 00 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 01 Side 0 -> nicht formatiert. Weitersuchen? (J/N) J
Cyl 01 Side 1 -> DD Track 01 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 02 Side 0 -> DD Track 01 Head 00 Size 2 Sek 00 08 03 06 01 09 04 07 02 05
Cyl 02 Side 1 -> DD Track 02 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 03 Side 0 -> nicht formatiert. Weitersuchen? (J/N) J
Cyl 03 Side 1 -> DD Track 03 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 04 Side 0 -> DD Track 02 Head 00 Size 2 Sek 00 08 03 06 01 09 04 07 02 05
Cyl 04 Side 1 -> DD Track 04 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 05 Side 0 -> nicht formatiert. Weitersuchen? (J/N) J
Cyl 05 Side 1 -> DD Track 05 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 06 Side 0 -> DD Track 03 Head 00 Size 2 Sek 00 08 03 06 01 09 04 07 02 05
Cyl 06 Side 1 -> DD Track 06 Head 01 Size 2 Sek 01 02 03 04 05 06 07 08
Cyl 07 Side 0 -> nicht formatiert. Weitersuchen? (J/N) N
Spurwechselzeit in ms? (1 bis 32) 4
```

**Bild 3. »Readid« sieht so aus**

mals länger sein, als die Sektorlänge zusammen mit dem Sektorabstand beim Formatieren. Sicherheitshalber muß gelten:

$$(ranf+dat+fgap)/(1+t) > (ranf+dat+rgap)*(1+t)$$

Dieselbe Bedingung gilt für »rmin«, die minimale Länge des Abstandes beim Formatieren.

Alle Sektoren müssen auch dann auf die Spur passen, wenn der Umfang der Diskette wegen Schwankungen in der Umdrehungsgeschwindigkeit um den Faktor $(1 + t)$ zu kurz erscheint. Für das maximale fgap gilt also die Bedingung:

$$(tanf+n*(fanf+ranf+dat+fgap)) < tlen/(1+t)$$

Für den $\mu$PD765 kann fgap maximal 255 Byte lang werden, selbst wenn die obige Bedingung einen größeren Wert liefert.

Es gilt also immer:
$fgap < 256$

Die neu geschriebenen Teile des Sektors zusammen mit der Pause beim Schreiben (Lesen), müssen selbst dann die Reste des Datenteils eines überschriebenen Sektors überdecken, wenn dieser um den Faktor $(1 + t)$ zu lang und gleichzeitig der neue Sektor um den Faktor $(1 + t)$ zu kurz ist. Für die minimale Länge der Lücke ergibt sich die folgende Bedingung:

$$(ranf+dat+rgap)/(1+t) > (ranf+dat)*(1+t)$$

Aus den letzten drei Bedingungen ergeben sich durch Auflösen direkte Grenzwerte für rgap und fgap:

$$rgap > (ranf+rdat)*((1+t)^2-1)$$
$$fgap < MINIMUM(255, (tlen/(1+t)-tanf)/n-(fanf+ranf+dat))$$

Bei einer vorgegebenen Toleranz des Laufwerks gibt es jetzt mehrere erlaubte Kombinationen von rgap und fgap. Im nächsten Schritt muß man die optimale Kombination herausfinden. Das ist diejenige, bei der die erlaubte Toleranz möglichst groß wird. Das Laufwerk sollte dann aber diese Toleranz nicht mehr ausnutzen.

Das Resultat all dieser Überlegungen ist das Programm »Gaps« (Listing). Das sehr einfache, aber hier ausreichende Näherungsverfahren für eine gegebene Sektorlänge und -zahl berechnet die optimalen Gaplängen und die maximal erlaubte Laufwerkstoleranz. Zuerst geben Sie die Sektorlänge, dann durch ein Leerzeichen getrennt, die Anzahl der Sektoren pro Spur, ein. Wenn das Programm keine Werte liefert, ist die eingetippte Kombination verboten. In diesem Fall müssen Sie eine geringere Sektorlänge oder Sektorzahl wählen.

Die Pausezeit »rgap« wird beim Schreiben von Sektoren dazu genutzt, den Schreib-/Lesekopf vom Schreib- in den Lesebetrieb umzuschalten. Wenn Ihr Laufwerk das in rgap*0,03 Millisekunden nicht schafft, müssen Sie einen Sektor der Spur weglassen. Im Zweifelsfall helfen (wie immer) Experimente.

## Sektormarkierungen lesen

Das Kommando »Readid« liest die vier Byte der beim Formatieren angegebenen Markierung des nächsten Sektors, auf den der Schreib-/Lesekopf trifft. So läßt sich zum Beispiel das Format einer Diskette bestimmen.

Ein Beispiel für dieses Kommando ist das Programm »Readid« (Seite 136). Das Ergebnis zeigt Bild 3. Es führt eine vollständige Diskettenanalyse durch, wobei es alle in Single Density oder in Double Density geschriebenen Sektoren ausfindig macht. Von den Informationen werden wegen der besseren

Übersicht nur Ausschnitte dargestellt. Bei besonders hartnäckigen Formaten schaffen Sie aber durch Einfügen von einigen »write«-Anweisungen leicht Abhilfe.

Sektoren können nur übersehen werden, wenn sich auf einer Spur mehrere Sektoren mit derselben Nummer befinden. Durch mehrmaliges Analysieren einer Spur und Weiterverwenden des Ergebnisses mit der größten Sektorenzahl lösen Sie aber auch dieses Problem.

## Schreiben und Lesen von Daten

Die Kommandos zum Schreiben oder Lesen von Daten bewegen den Schreib-/Lesekopf nicht. Bearbeitet wird immer der Zylinder, auf dem sich der Kopf gerade befindet. Die Bytes C, H, R und N entsprechen den ID-Bytes des ersten gesuchten Sektors. Wenn diese Sektoren 128 Byte lang sind (N=0), kann man mit dem Byte DTL angeben, wieviele Bytes jedes Sektors übertragen werden sollen. GPL ist dabei die rgap, die im letzten Abschnitt berechnet wurde.

Alle Schreib- und Lesekommandos übertragen automatisch Daten mehrerer aufeinanderfolgender Sektoren. Die Übertragung endet erst dann, wenn der Controller entweder von einem Peripheriegerät ein »Terminal-Count-Signal« erhält oder der letzte Sektor der Spur übertragen wurde. Bei den Schneider-Computern kann kein Terminal-Count-Signal erzeugt werden. Deshalb muß im Kommandobyte »EOT« nicht der letzte Sektor der Spur, sondern der letzte zu übertragende Sektor angegeben werden. Wenn nur ein Sektor bearbeitet werden soll, muß also das Byte R denselben Wert erhalten wie EOT. Diese Methode hat aber den Fehler, daß auch bei einem korrekten Übertrag ein Fehler gemeldet wird.

## Zylinderweise

Wenn das »Multi-Track-Bit« gesetzt ist, endet die Übertragung nicht am Sektor EOT auf der Seite Null, sondern sie wird beim Sektor Nummer 1 auf der Diskettenseite 1 fortgesetzt. Erst wenn der Zylinder vollständig gelesen ist, endet das Kommando. Mit einer einzigen Anweisung können so bei Disketten mit 250000 Hz Aufzeichnungsfrequenz bis zu 10 KByte Daten übertragen werden.

Die Sektoren enthalten außer den ID-Bytes einige zusätzliche Markierungen, die nicht unmittelbar zu lesen sind. Eine dieser Markierungen hat den etwas irreführenden Namen »Deleted«. Beim Schreiben von Daten wird angegeben, ob die betroffenen Sektoren die Markierung »Deleted« oder »Not Deleted« erhalten sollen. Mit dem Bit »SK« wird entschieden, wie die Markierungen beim Lesen behandelt werden. Wenn beim Lesen von als »Not Deleted« markierten Daten das SK-Bit gesetzt ist, werden mit »Deleted« markierte Daten einfach übersprungen. Wenn dieses Bit nicht gesetzt ist, wird der erste als »Deleted« markierte Sektor zwar gelesen, das Kommando aber sofort danach abgebrochen. Beim Lesen von mit »Deleted« markierten Daten hat das SK-Bit die umgekehrte Bedeutung. Bei gesetztem Bit werden jetzt »Not Deleted«-Daten übersprungen und bei nicht gesetztem Bit entsprechend dem Kommando bei »Not Deleted«-Daten abgebrochen.

Bei den Kommandos »Read Data«, »Read Deleted Data«, »Write Data« und »Write Deleted Data« werden Sektoren in der Reihenfolge ihrer Numerierung übertragen, also unabhängig von der Reihenfolge auf der Diskette. Eine Ausnahme ist das Kommando »Read A Track«. Bei diesem werden die Datenbytes der Sektoren, unabhängig von der Numerierung, in der Reihenfolge übertragen, in der sie auf der Diskette stehen. Die Übertragung beginnt bei »Read A Track« mit dem Sektor, der sich unmittelbar hinter dem Indexloch der Diskette befindet. Statt des letzten Sektors wird hier die Anzahl der zu übertragenden Bereiche angegeben. Bei diesem Befehl bricht die Datenübertragung auch dann nicht ab, wenn ein Lesefehler auftritt. Falls die Sektoren auf der Diskette nicht aufsteigend numeriert sind oder der erste Sektor eine andere Nummer als die angegebene hat, läuft die Datenübertragung ebenfalls weiter. In der Ergebnisphase werden allerdings Fehler gemeldet.

In dieser Phase werden die ID-Bytes desjenigen Sektors zurückgegeben,

```
program GAP-Laengenberechnung;
const tlen: real = 6250; tanf: real = 146; ranf: real = 40; fanf: real = 22;
      gen: real = 0.00001;
var   slen, ns, rges, fgap, rgap, st, t, t1, t2: real;
begin repeat
  writeln; write('Sektorlaenge, Anzahl: '); readln(slen, ns);
  writeln('Toleranz GAP-Format GAP-Read/Write');
  rges := ranf + slen; t := 0;
  st := 0.5; repeat
    t1 := t+1; t2 := t1*t1;
    fgap := (tlen/t1-tanf)/ns-(fanf+rges); if fgap >= 255.5 then fgap := 255;
    rgap:=(t2-1)*rges;
    if ((rges+fgap) > ((rges+rgap)*t2))
      then begin writeln('':2,t:6:4,'':6,fgap:3:0,'':12,rgap:3:0); t:=t+st end
      else if t >= st then t := t-st;
  st := st/2 until st < gen
until false end.
```

**Listing. Wie lang ist der »Gap«?**

| Vorgeschlagene Werte für GPL bei verschiedenen Formaten | | | | | |
|---|---|---|---|---|---|
| Density | Sektor-größe | Größen-kennung | Sektor-anzahl | GPL bei Read/Write | GPL beim Formatieren |
| single | 128 | 00hex | 18hex | 07hex | 09hex |
| single | 128 | 00hex | 10hex | 10hex | 19hex |
| single | 256 | 01hex | 08hex | 18hex | 30hex |
| single | 512 | 02hex | 04hex | 46hex | 87hex |
| single | 1024 | 03hex | 02hex | C8hex | FFhex |
| single | 2048 | 04hex | 01hex | C8hex | FFhex |
| double | 256 | 01hex | 12hex | 0Ahex | 0Chex |
| double | 256 | 01hex | 10hex | 20hex | 32hex |
| double | 512 | 02hex | 08hex | 2Ahex | 50hex |
| double | 1024 | 03hex | 04hex | 80hex | F0hex |
| double | 2048 | 04hex | 02hex | C8hex | FFhex |
| double | 4096 | 05hex | 01hex | C8hex | FFhex |

**Tabelle. Werte für GPC bei verschiedenen Formaten**







der dem letzten übertragenen logisch folgen würde.

Die Kommandos »Scan Equal«, »Scan Low Or Equal« und »Scan High Or Equal« arbeiten sehr ähnlich wie die Kommandos zum Schreiben und Lesen von Daten. Hier vergleicht der Controller aber Daten, die von den Laufwerken geliefert werden mit Daten, die der Computer liefert. Die Operation wird beendet, sobald ein Byte gefunden ist, das die gewünschte Eigenschaft erfüllt oder wenn das Kommando durch ein Terminal-Count-Signal oder Erreichen des letzten Sektors abgebrochen wird. Bei »Scan Low Or Equal« dauert die Suche damit höchstens solange, bis das Laufwerk ein Byte liefert, dessen Wert kleiner als der des gerade von der CPU gelieferten Bytes ist. Wenn bei der Suche alle Bytes identisch waren, wird in den Statusbits der Ergebnisphase unabhängig vom Kommando das Bit »Scan Equal Hit« gesetzt. Wenn weder ein Byte gefunden wird, das die Bedingung erfüllt, noch alle Bytes identisch sind, ist das Bit »Scan Not Satisfied« gesetzt. Nur wenn ein Wert gefunden wurde, der die »Scan«-Bedingung erfüllt, ist keines der beiden Bits gesetzt.

## Die SCAN-Kommandos

Das Byte »STP« gibt an, um wieviel die aktuelle Sektornummer nach dem Untersuchen des aktuellen Sektors erhöht wird. Das Resultat ist die Nummer des als nächstes zu untersuchenden Sektors. Wenn STP den Wert 1 besitzt, werden wie beim normalen Zugriff aufeinanderfolgende Sektoren untersucht; hat STP den Wert 2, nur jeder zweite Sektor. Auf diese Weise kann ein eventueller softwaremäßiger Versatz nachvollzogen werden. Zu beachten ist allerdings, daß der Sektor mit der Nummer EOT tatsächlich unter den zu untersuchenden Daten ist. Wenn STP so eingestellt ist, daß nur ungeradzahlige Sektoren untersucht werden, muß also auch EOT einen ungeradzahligen Wert enthalten.

Sie wissen jetzt alles, was das Programmieren Ihres Diskettenlaufwerks voraussetzt. Jetzt steht eigenen Anwendungen nichts mehr im Wege, zum Beispiel mit einem Kopierschutz Marke Eigenbau. Oder einem Programm, das mit Ihrem Original-Schneider-Controller beliebige fremde Diskettenformate lesen kann, einschließlich doppelseitigen Disketten. Vorstellbar ist auch ein superschnelles Kopierprogramm, das mit Hilfe von »Readid« feststellt, welche Sektoren als nächste vorbeikommen und diese dann ohne Wartezeit liest. Vielleicht wagen Sie sich auch an ein neues und schnelleres CP/M-BIOS.

(Helmut Tischer/hg)

# Freie Auswahl mit Format

**Jeder Computer beschreibt seine Disketten in einem ihm eigenen Format. Mit nur wenigen »Pokes« kann man aber seinem Schneider CPC beibringen, auch fremde Daten zu lesen.**

Dem Betriebssystem CP/M wird nachgesagt, es sei das mit dem größten Software-Angebot. Das ist richtig, aber was hilft es, wenn Ihr Computer die Programme von fremden Geräten nicht lesen kann. Anders als unter MS-DOS gibt es nämlich für CP/M kein standardisiertes Aufzeichnungsverfahren. Aber nicht verzagen, mit nur wenigen Anweisungen kann auch der Schneider- (und auch der Vortex-) Controller fast alle fremden Datenträger lesen.

Allen Diskettenformaten gemeinsam ist die Aufteilung in Spuren und Sektoren. Eine Diskette ist dabei in 40 oder 80 Spuren aufgeteilt, die sich wiederum in einzelne (normalerweise 8) Sektoren gliedern. Das Indexloch dient dabei dem Controller zur Orientierung, um sich auf der Diskette zurechtzufinden (siehe Bild 1).

Die grundsätzliche Struktur einer 3-Zoll-Diskette sieht übrigens genauso aus wie die einer Diskette mit 3½- oder 5¼-Zoll-Format. Deshalb darf man die 3-Zoll-Laufwerke der Schneider-Computer ohne weiteres durch Stationen für andere Diskettengrößen ersetzen. Da fast alle CP/M-Programme auf 5¼-Zoll-Disketten vorliegen, ist es sinnvoll, seinen Schneider mit einem 5¼-Zoll-Zweitlaufwerk auszustatten.

Grundsätzlich kennt der Disketten-Controller von Schneider zwei verschiedene Aufzeichnungsverfahren: das »IBM 3740 Single Density«- und das »IBM System 34 Double Density«-Format. Diese beiden Formate sind seit ihrer Einführung in den 70er Jahren so weit verbreitet, daß es sich kaum ein Hersteller leisten kann, seinem Computer ein eigenes Aufzeichnungsverfah-

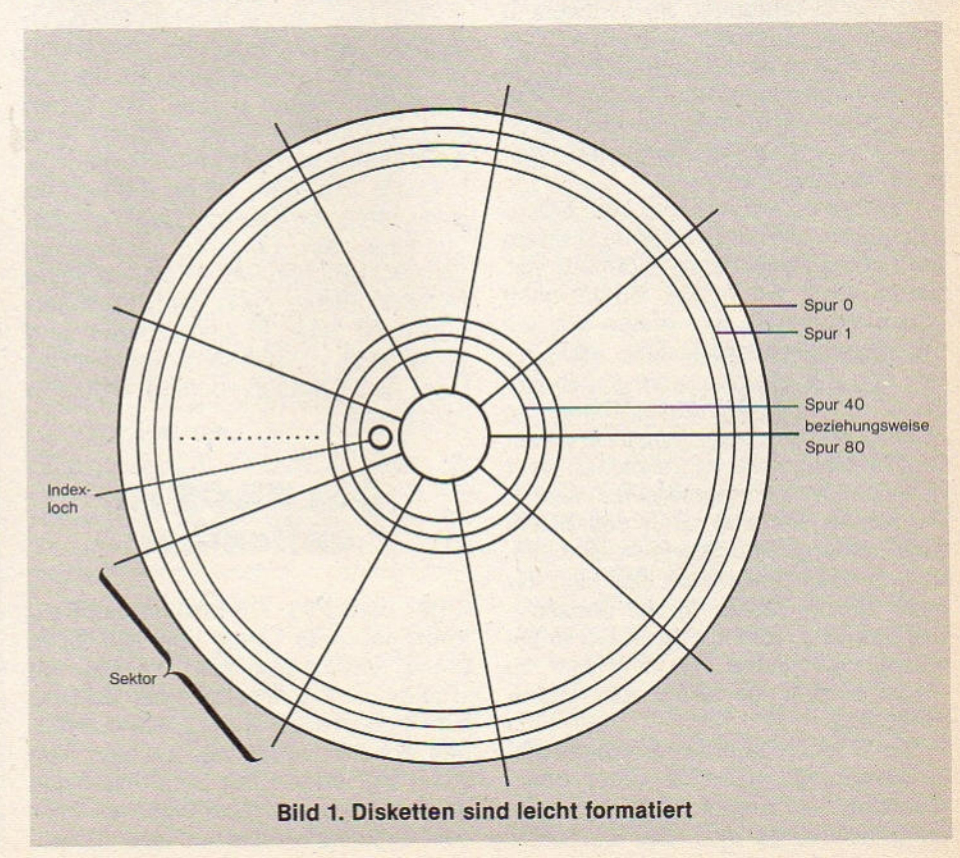



**Bild 1. Disketten sind leicht formatiert**

ren zu verpassen. Allein Apple und Commodore arbeiten mit einem eigenen Format. Commodore allerdings greift beim Amiga ebenfalls auf das bewährte Datenformat von IBM zurück.

Jedes der beiden IBM-Formate gibt es mit einer Aufzeichnungsfrequenz von 250 und 500 kHz. Diese Frequenz darf man nicht mit der Bitrate verwechseln, mit der die Daten dann tatsächlich geschrieben beziehungsweise gelesen werden. Die höhere Geschwindigkeit von 500 kHz kommt fast ausschließlich bei 8-Zoll-Laufwerken zur Anwendung. Die kleinen Formate arbeiten mit der niedrigeren Geschwindigkeit, so auch Schneider und Vortex.

Beim Übertragen von CP/M-Programmen treten somit in aller Regel keine Probleme mit dem Aufzeichnungsverfahren auf.

## Wo steht was?

Anders sieht es mit dem Platz aus, an dem sich die Daten auf der Diskette befinden. Unter CP/M existieren dazu die unterschiedlichsten Formate. Am leichtesten erkennen Sie diese an den verschiedenen Speicherkapazitäten und an den unterschiedlich großen Inhaltsverzeichnissen. Selbst ein einziger CP/M-Computer arbeitet oft mit verschiedenen Formaten – beispielsweise eins für eine Festplatte und ein anderes für die Disketten. Beim Schneider ändert sich das Format sogar beim Diskettenwechsel. So gibt es das System- und das Datenformat. Im Laufwerk B erkennt der Controller auch unter CP/M dieses Format ohne Systemspuren. Da die Betriebssystem-Routinen im BDOS die gleichen sind, muß man also durch unterschiedliche Parameter den Controller umstellen können.

Dazu ist es erforderlich, dem BDOS mitzuteilen, wie die zu lesende Diskette (im Rahmen bestimmter Normen) verwaltet wird. Wenn das BDOS nach einem Warmstart zum ersten Mal ein Diskettenlaufwerk anspricht, dann liefert der vom Computertyp abhängige BIOS-Teil des Betriebssystems die Anfangsadresse einer Tabelle. In dieser Tabelle stehen die Informationen über das Format der eingelegten Diskette. Danach richtet sich dann das BDOS beim späteren Datentransfer. Unmittelbar vor einem Warmstart darf sich der Inhalt dieser Tabelle immer ändern – und das kann nicht nur das BIOS selbst veranlassen. Auch Sie sind hier zu einem Eingriff ins BIOS »per Hand« berechtigt.

Zum Beschreiben dieser Tabelle gibt es verschiedene Wege. Unter Basic steht dafür der Befehl POKE zur Verfügung (allerdings muß das ein unter CP/M laufender Interpreter sein – zum Beispiel C- oder MBasic). Unter Turbo-Pascal dient dazu das Pseudo-Bytefeld »mem(..)« oder eine Variablendefinition mit »Absolute«. Der Debugger DDT kennt hierfür den »S«-Befehl. Haben wir damit die Lösung unserer Probleme gefunden?

Leider sind die Regeln für die Verwaltung von CP/M-Disketten nicht vollständig. Für einige Details gibt es verschiedene erlaubte Lösungen. Und – wie sollte es anders sein – jeder Hersteller wählt eine andere.

Ein Beispiel dafür sind doppelseitig beschriebene Disketten, wie sie beispielsweise der Vortex-Controller produziert. Das Standard-CP/M kennt nur eine Diskettenseite pro Laufwerk. Deshalb unterscheiden sich die Spuren auf der Oberseite von den Spuren auf der Unterseite durch die Spurnummer. Die Umrechnung der von CP/M benutzten (in die der Hardware bekannten) Kennziffern übernimmt das vom Computertyp abhängige BIOS des Betriebssystems.

Und da treten Probleme auf. Bei manchen Systemen liegen alle Spuren mit geradzahligen Nummern auf der Oberseite der Diskette und ungeradzahlige auf der Unterseite. Bei anderen tragen die oberen Spuren die Ziffern Null bis 79 und die unteren 80 bis 159. Bei dritten sind die Spuren auf der Unterseite in der umgekehrten Reihenfolge angeordnet. Die niedrigste Kennziffer bezeichnet die Spur im Zentrum und die höchste die am Rand.

Beim Datenzugriff müssen Sie daran denken, daß die Spur- und Sektornummern, die beim Formatieren auf die Diskette geschrieben wurden, beim Lesen wieder zurückgegeben werden. Und auch hierbei gibt es zwei verschiedene Formate. Entweder haben die sich an einer Stelle der Diskette direkt gegenüberliegenden Sektoren unabhängig von der logischen Reihenfolge dieselbe Nummer (diese heißt dann »Zylindernummer«) oder in den Spuren sind unabhängig von der Seite die logischen (und damit verschiedenen) Sektoren eingetragen.

## Das BDOS ist unflexibel

Das BDOS von CP/M kennt nur Sektoren mit einer Länge von 128 Byte. Diese BDOS-Sektoren werden oft »Records« genannt. Auf einer Diskette sind jedoch auch andere Sektorenlängen erlaubt. Heutzutage benutzt man meist 512 oder 1024 Byte. Allerdings wird ein Sektor auf einer Diskette immer am Stück und nicht in mehreren Portionen gelesen und geschrieben. Das BIOS des Computers muß also die BDOS-Sektoren selbständig zu größeren Paketen bündeln beziehungsweise auseinanderdividieren. Dazu bekommt es vom BDOS nur wenige Hilfen.

Das BIOS schreibt einen bearbeiteten Sektor in der Regel nur dann auf Diskette, wenn der Sektorpuffer für andere Aufgaben benötigt wird. Falls mehrere Records hintereinander in denselben Sektor geschrieben werden, spart man somit einige Diskettenzugriffe ein. Das bringt aber auch eine große Gefahr: Falls unmittelbar nach dem Schreiben (und damit vor dem physikalischen Datenübertrag) das System durch einen Warmstart neu gestartet wird, geht der letzte Sektor verloren. Das BDOS setzt deshalb beim Schreiben ins C-Register die Zahl 1, wenn der Sektor ausnahmsweise auch physikalisch sofort geschrieben werden soll. Umgekehrt liest das BIOS auch zuerst jeden Sektor von der Diskette, bevor einzelne Records durch neue ersetzt werden. Wenn der Sektor vorher leer ist, erübrigt sich das allerdings. In solch einem Fall meldet das BDOS im C-Register den Wert 2. Nur beim »normalen« Diskettenzugriff steht im C-Register der Wert 0.

## Physik kontra Logik

Wie weiter oben ausgeführt, haben also die physikalischen Sektoren auf einer Diskette andere Nummern als die logischen Sektoren. Beim Systemformat von Schneider tragen die Sektoren zum Beispiel Nummern zwischen 41 und 49 hex, beim Datenformat hingegen Ziffern zwischen C1 und C9 hex. Das IBM- und das Vortex-Format zählen von 01 hex an aufwärts. Und da das BIOS sowieso schon beim Rechnen ist, wird oft auch noch die Reihenfolge der Sektorennummern geändert. Der beim Umrechnen erzeugte Sektorversatz darf übrigens nicht mit dem Sektorversatz beim Formatieren verwechselt werden. Manchmal sind auch noch die beiden Spuren eines Zylinders als eine logische Spur bezeichnet, und die Sektoren werden zwischen beiden »wild« aufgeteilt.

Das ist aber noch nicht alles. Einige wenige Formate arbeiten mit einer »invertierten« Aufzeichnung, bei der in allen Datenbytes die Bits mit dem Wert 1 zurück- und die Bits mit dem Wert 0 gesetzt werden. Aus der Bitfolge »01001110« für den Buchstaben »N« wird dann »10110001«. Falls eine Diskette mit 40 Spuren in einem Laufwerk mit der doppelten Spurzahl liegt, muß beim Lesen jede zweite Spur übersprungen werden. Bei bestimmten

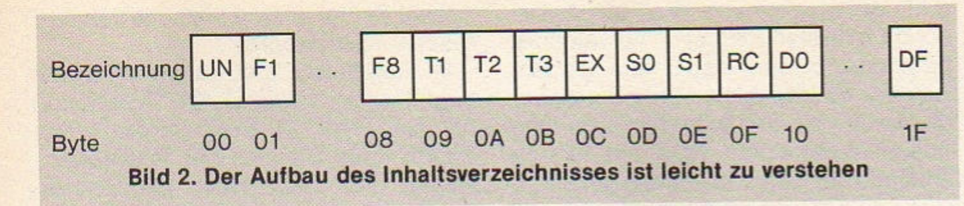
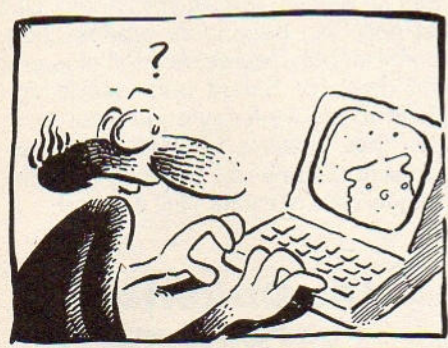
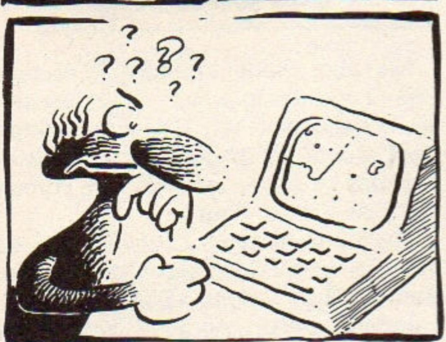


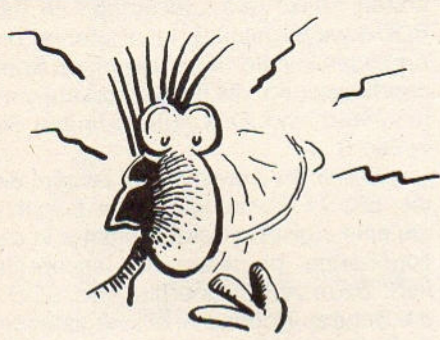
**Bild 2. Der Aufbau des Inhaltsverzeichnisses ist leicht zu verstehen**

Operationen auf der Diskette ist sogar der Abstand zwischen zwei einzelnen Sektoren von Bedeutung.

Bei soviel Wirrwarr scheint es unmöglich, Disketten in einem fremden Format mit dem Schneider lesen zu können. Doch keine Angst! Alle Inkompatibilitäten bei nicht standardisierten Details der Diskettenformate lassen sich auf eine einzige »zentrale Umrechnungsroutine« reduzieren. Diese rechnet Laufwerks-, Spur- und Recordnummer in Geräte-, Zylinder- und Kopfadresse um. Gleichzeitig werden die formatierte Spur- und Kopfnummer sowie Sektorgröße, -nummer und Lage des Records im Sektor bestimmt. Darüber hinaus muß das BIOS dann nur noch zwischen invertierter und nicht invertierter Aufzeichnung sowie zwischen 40 (Single) und 80 Spuren (Double Density) unterscheiden.










Das BIOS einiger Computertypen unterstützt nur eine einzige Umrechnungsvariante der gelieferten Daten. Andere BIOS-Versionen besitzen analog zu der dem BDOS übergebenen Parametertabelle intern eine zweite Tabelle für die Diskettenstation. In dieser werden dann verschiedene Parameter der Umrechnung eingestellt. Diese »BIOS-Tabelle« steht im RAM des Computers, so daß sie leicht zu ändern ist. Je leistungsfähiger ein BIOS, desto besser läßt sich die Umrechnung steuern. Das BIOS der Schneider-Computer zeigt sich sehr anwenderfreundlich. Es läßt die Voreinstellung fast aller Parameter zu, soweit das überhaupt sinnvoll erscheint. Programme, die unterschiedliche CP/M-Formate lesen, verändern diese Tabelle. Für Besitzer einer Vortex-Disketten-Station gibt es dazu »Para« (das Programm ist aber leider nicht ganz fehlerfrei – an einer verbesserten Version wird gearbeitet). CP/M Plus-Anwender finden auf Seite 122 in Happy-Computer, Ausgabe 12/86, ein geeignetes Programm.

Im folgenden beschränken wir uns auf CPM 2.2. Zum einen gibt es noch kein passendes Programm für den Schneider-Controller unter CP/M Plus, und zum zweiten ist auf den Systemdisketten für den CPC 6128 auch das ältere Betriebssystem zu finden. Das Umsetzen auf CP/M Plus bleibt dann Ihrem eigenen Erfindungsgeist vorbehalten.

Unter CP/M wird eine Diskette in einen System- und einen Datenbereich aufgeteilt. Für den Systembereich sind die äußersten zwei Spuren der Diskette reserviert. Es dürfen aber auch weniger oder mehr sein. Auch Disketten ganz ohne Systemspuren sind erlaubt. Diese (normalerweise) zwei Spuren benutzt das eigentliche CP/M nicht. Das BIOS darf sie für individuelle Informationen gebrauchen.

Der Datenbereich einer Diskette ist in Blöcke der Länge 2, 4, 8 oder 16 KByte aufgeteilt. Bei Disketten mit einem Datenbereich von maximal 256 KByte ist auch die Blocklänge 1 KByte erlaubt. Blockgrenzen brauchen dabei übrigens nicht mit den Spurgrenzen übereinzustimmen. Eine Datei beginnt immer an einer Blockgrenze. Selbst wenn sie nur wenige Bytes lang ist, wird immer ein voller Block reserviert. Der restliche Raum des Blockes bleibt ungenutzt. Somit ist es günstig, ein Format mit möglichst kleiner Blocklänge zu verwenden.

Der erste Block des Datenbereiches trägt die Nummer Null und ist immer für das Inhaltsverzeichnis reserviert. Falls ein Block nicht ausreicht, werden noch die nächstfolgenden Blöcke freigehalten. Ein einzelner Eintrag im Inhaltsverzeichnis belegt immer 32 Byte (siehe Bild 2).

Im ersten Byte steht die Usernummer, zu der der Verzeichniseintrag gehört. Erlaubt sind die Werte Null bis 15. Der Wert E5 hex markiert einen gelöschten Eintrag. Diesen überschreibt beim nächsten Beschreiben der Diskette die neue Datei. Die Bytes F1 bis F8 und T1 bis T3 enthalten den Dateinamen zuzüglich Erweiterung. Nicht benutzte Bytes werden mit Leerzeichen aufgefüllt. Da ASCII-Zeichen mit nur 7 Bit eindeutig gekennzeichnet sind, benutzt CP/M das 8. Bit für spezielle interne Informationen. Wenn das 8. Bit des Byte T1 gesetzt ist, wird die Datei von BDOS-Operationen nicht mehr beschrieben (R/O-Datei). Das gesetzte 8. Bit von T2 bewirkt, daß die Datei beim Befehl »DIR« nicht ausgegeben wird (SYS-Datei). Die 8. Bits der Bytes F1 bis F4 stehen für beliebige Informationen zur Verfügung. Alle übrigen Bits (also von F5 bis F8 und T3) sind späteren Versionen des Betriebssystems CP/M vorbehalten.

## Ab 16 Blöcke wird es kompliziert

In den Bytes D0 bis DF stehen die Kennziffern der Blöcke, die für die Datei reserviert sind. Falls der Datenbereich der Diskette maximal 256 Blöcke kennt, sind die Blocknummern 8 Bit lang, und es können 16 Blöcke reserviert werden. Wenn der Datenbereich eine größere Zahl von Blöcken enthält, ist eine Blocknummer 16 Bit lang, und es können maximal 8 Blöcke reserviert werden. Hierbei tritt nun ein Problem auf. Unter CP/M 2.2 darf eine Datei theoretisch bis zu 8 MByte lang werden. Selbst wenn die Blöcke maximale

Länge haben, können in einem Eintrag nur 256 (=16*16) KByte reserviert werden. Falls eine Datei mehr als 16 (beziehungsweise 8) Blöcke enthält, wird in das Inhaltsverzeichnis ganz einfach ein weiterer Eintrag mit demselben Dateinamen geschrieben. Darin stehen dann die zusätzlichen Blöcke für die Datei. Falls auch dieser nicht ausreicht, so wird ein dritter angelegt.

In den Bytes EX, S1, S2 und RC wird die Reihenfolge der Verzeichniseinträge und die Anzahl der Records, die im aktuellen Eintrag belegt sind, geregelt. Beide Informationen sind in einer einzigen Zahl verschlüsselt.

Eine Datei darf maximal 65535 Records belegen. Die Kennziffern beginnen mit Null. Im Inhaltsverzeichnis wird in den Bytes EX, S1, S2 und RC die Kennziffer des letzten Records plus 1 vermerkt. Falls ein Eintrag für die Datei nicht ausreicht, steht hier der höchste Record dieses Eintrags plus 1. Dieser Wert entspricht damit der Nummer des ersten Records, der nicht mehr in diesen Eintrag paßt, und für den ein weiterer Eintrag reserviert ist.

Ein Beispiel: Falls eine Datei 1200 Records belegt, im Inhaltsverzeichnis pro Eintrag allerdings nur Platz für 512 Records reserviert ist, dann belegt die Datei drei Einträge. Der erste ist vollständig gefüllt. Der letzte hiermit angesprochene Record hat die Nummer 511. In den 4 Byte EX, S0, S1 und RC ist also der Wert 512 eingetragen. Der zweite Eintrag ist ebenfalls vollständig gefüllt, wobei der Record mit der Nummer 1023 als letzter eingetragen ist. In den 4 Byte steht deshalb der Wert 1024. Der letzte Record der zu schreibenden Datei hat die Nummer 1199, weshalb im dritten Eintrag im Verzeichnis der Wert 1200 steht.

## Eintrag der Recordnummer

Weil das alles so aber noch zu einfach ist, werden die 16 Bit der Recordnummer nicht einfach als Integerzahl abgelegt. Die Bits Null bis 6 der Recordnummer stehen im Byte RC, die Bits 7 bis 11 in EX und die Bits 12 bis 15 in S1. Das Byte S0, die Bits 5 bis 7 von EX und die Bits 4 bis 7 von S1 werden nicht benutzt und haben immer den Wert Null. Auch Bit 7 des Byte RC muß nach dem bisher Besprochenen immer den Wert Null haben.

Auch hat die Regel bis jetzt noch einen schweren Mangel. Falls nämlich die Datei aus dem Beispiel genau 512 Records lang ist, so wird im zweiten Eintrag der erste nicht mehr belegte Record eingetragen – also der Wert 512. Genau derselbe Wert steht aber schon im ersten Eintrag im Verzeichnis. Es ist nun nicht mehr zu unterscheiden, welcher Eintrag der erste und welcher der zweite ist. Wir brauchen also eine Ausnahmeregelung: Falls ein Eintrag am Ende der Datei vollständig gefüllt ist, wird von der eigentlich einzutragenden Kennziffer der Wert 128 abgezogen und das neue Ergebnis eingetragen. Für eine Datei mit 512 Blöcken ist das dann die Ziffer 384. Das gesetzte Bit 7 von RC weist auf diesen Ausnahmezustand hin. Zur Kennzeichnung dieses Ausnahmezustandes wird im Byte RC das Bit Nr. 7 gesetzt.

## Experimente mit dem Inhaltsverzeichnis

Falls die letzten Erklärungen etwas schnell an Ihnen »vorbeigerauscht« sind, lassen Sie sich keine grauen Haare wachsen. Das Programm aus Listing 1 gibt das vollständige Inhaltsverzeichnis der Diskette im Bezugslaufwerk aus. Aus den schon auf Seite 129 erklärten Gründen ist es in Turbo-Pascal geschrieben. Wer keinen Turbo-Pascal-Compiler besitzt, dem ist mit der Leserservice-Diskette geholfen. Auf ihr befinden sich alle Programme als lauffähige COM-Dateien. Mit Hilfe dieses Programms oder eines Diskettenmonitors betrachten Sie die Einträge verschieden langer Dateien. Vieles wird dann sofort klarer. Hilfreich ist übrigens, wenn Sie in Gedanken eine Datei aufbauen und nach jedem Record die Verzeichniseinträge und die darin stehenden Nummern aufschreiben.

Unabhängig vom Aufzeichnungsformat werden mindestens 16 KByte Daten pro Eintrag reserviert. Damit nun nie Daten verlorengehen, schließt das BDOS vorsichtshalber auch bei längeren Einträgen nach 16 KByte die Datei. Ein Eintrag im Verzeichnis wird in der Terminologie von CP/M »physikalischer Extent« genannt. Entsprechend heißt ein 16 KByte langer Dateiabschnitt »logischer Extent«. Ein logischer Extent enthält damit also bis zu 128 Records. In den Bytes EX und S1 steht also immer die Nummer des letzten reservierten logischen Extents des physikalischen Extents. Das Byte RC enthält die Anzahl der im letzten logischen Extent reservierten Records.

Im Gegensatz zum wahlfreien Dateizugriff mit speziellen BDOS-Funktionen hat das Byte S1 beim sequentiellen Zugriff (zum Beispiel bei Textdateien) immer den Wert Null. Sequentielle Dateien können damit maximal 512 KByte lang werden. Byte S1 berücksichtigen viele Programme deshalb gar nicht erst. Byte EX heißt dann »das Extentbyte«. Eine CP/M-Datei kann übrigens auch »Löcher« enthalten: Ein nicht belegter Block mitten in einer Datei ist im Verzeichnis mit Null gekennzeichnet. Ein Eintrag, der nur unbelegte Blöcke enthält, entfällt vollständig. Das ist dann ein besonders großes Loch. Somit ist beispielsweise auch eine erst mit dem 16. Extent beginnende und dann normal weiterlaufende Datei denkbar. Das Inhaltsverzeichnis zeigt allerdings nur Dateien an, bei denen der Eintrag mit der Nummer Null existiert.

## Die Diskettenverwaltung

Nun aber wieder zurück zum Ändern des Diskettenformates. Sie wissen jetzt, nach welchen Gesichtspunkten der Datenträger eingeteilt ist und inwieweit das Inhaltsverzeichnis vom Format der Diskette abhängt.

Für jedes einzelne Laufwerk gibt es im BIOS des CP/M eine zentrale Tabelle, den 16 Byte langen »Disk Parameter Header«. Deren Adresse wird beim ersten Aufruf des Laufwerkes an das BDOS weitergeleitet. In ihr stehen die Adressen von den verschiedenen Speicherbereichen, die für dieses Laufwerk reserviert sind. Den Aufbau finden Sie in Bild 3.

»XLT« ist die Adresse der Tabelle, die die BIOS-Routine »Sectran« benutzt, um eine logische Sektornummer in die zugehörige physikalische umzurechnen. Beim Aufruf übergibt das BDOS der Sectran-Routine im BC-Register die logische Sektornummer und im DE-Register die Adresse der in Frage kommenden Tabelle. Im HL-Register wird die physikalische Sektornummer zurückgegeben. Bei Schneider wird die Sektornummer jedoch immer unverändert zurückgegeben. Die Sectran-Routine bleibt unbeachtet. Somit ent-

| Bezeichnung | XLT | 0000 | 0000 | 0000 | DIRBUF | DPB | CSV | ALV |
|---|---|---|---|---|---|---|---|---|
| Byte | 00-01 | 02-03 | 04-05 | 06-07 | 08-09 | 0A-0B | 0C-0D | 0E-0F |

**Bild 3. Der Disketten-Parameter-Header enthüllt alle Geheimnisse**

```
program dirleser;
(* Liest das vollstaendige
Inhaltsverzeichnis und zeigt es an *)

type datei=string[11];
var name:datei;
    dma :array[0..127]of byte;
    fcb :array[0..35]of byte;
    rc,i:integer;

procedure nb(n:integer);
begin
  if n>9 then
    write(chr(ord('A')-10+n))
  else
    write(chr(ord('0')+n))
end;

procedure wrhex(b:integer);
begin
  nb(b shr 4);
  nb(b and 15);
  write(' ')
end;

begin
  writeln('Ausgeben der gesamten
  Inhaltsverzeichnisinformation');
  writeln;
  bdos(26,addr(dma));
  fillchar(fcb,16,ord('?'));
  rc:=bdos(17,addr(fcb));
  while rc<>255 do
  begin
    rc:=rc shl 5;
    wrhex(dma[rc]);
    name:=ptr(addr(dma)+rc);
    name^[0]:=#11;
    write(name^,' ');
    for i:= rc + 12 to rc + 31 do
      wrhex(dma[i]);
    writeln;
    rc:=bdos(18)
  end;
end.
```

**Listing 1. Lesen Sie das Inhaltsverzeichnis Ihrer Disketten**

```
program getdisk;
(*Feststellen der Adressen und Inhalte
der Diskparameterbloecke *)

var olddisk           :byte;
    dphstart,freistart:integer;
    csvab,alvab       :array[0..1]of integer;

type bstr=string[2];
     wstr=string[4];

function nibble(n:integer):char;
begin
  if n>9 then
    nibble:=chr(ord('A')-10+n)
  else
    nibble:=chr(ord('0')+n)
end;

function convhexb(b:integer):bstr;
begin
  convhexb:=nibble(b shr 4)+nibble(b and 15)
end;

function convhexw(w:integer):wstr;
begin
  convhexw:=convhexb(w shr 8)+convhexb(w and 255)
end;

procedure showdpb(st:integer);
var i:integer;
begin
  write('dpb:');
  for i:=st to st+14 do write(' $',convhexb(mem[i]));
  writeln;
  write('dpx:');
  for i:=st+15 to st+24 do write(' $',convhexb(mem[i]));
  writeln
end;

begin
 olddisk:=bdos(25);
 dphstart:=bioshl(8,0);
 bdos(14,olddisk);
 if dphstart<$C000 then
  freistart:=$bd3a
 else
  freistart:=dphstart+$5c;
 csvab[0] := mem[dphstart + $0c] +
mem[dphstart + $0d] shl 8;
 csvab[1] := mem[dphstart + $1c] +
 mem [dphstart + $1d] shl 8;
 alvab[0] := mem[dphstart + $0e] +
 mem[dphstart + $0f] shl 8;
 alvab[1] := mem[dphstart + $1e] +
 mem[dphstart + $1f] shl 8;
 writeln('Achtung! Werte gelten nur fuer
CP/M 2.2 und nicht fuer CP/M Plus!');
 writeln('dphstart: $',convhexw(dphstart),
' freistart: $',convhexw(freistart));
 writeln('csvab: $',convhexw(csvab[0]),'
 und $',convhexw(csvab[1]));
 writeln('alvab: $',convhexw(alvab[0]),'
 und $',convhexw(alvab[1]));
 writeln('Laufwerk  A:');showdpb(mem[dphstart+$0a]
+mem[dphstart+$0b]shl 8);
 writeln('Laufwerk  B:');showdpb(mem[dphstart+$1a]
+mem[dphstart+$1b]shl 8)
 end.
```

**Listing 2. Manipulationen sind einfach**

halten die Bytes XLT immer den Wert 0. Die folgenden Bytes »0000« werden vom BDOS zum Speichern verschiedener Werte benutzt. Ihr Ausgangswert ist somit beliebig.

»Dirbuf« enthält die Adresse eines 128 Byte langen Speicherbereichs, den das BDOS für Manipulationen im Inhaltverzeichnis benötigt. Da immer nur ein Laufwerk gleichzeitig angesprochen wird, gibt es nur eine gemeinsame Adresse für alle Laufwerke.

»DPH« enthält die Adresse des Disketten-Parameter-Blocks des aktiven Laufwerks. Dieser Block bestimmt das genaue Format der Diskette. Für jedes Diskettenformat ist im System ein eigener Parameter-Block vorgesehen.

Die Informationen über das Format der Diskette werden allerdings nur beim ersten Zugriff gelesen. Nach einem unangemeldeten Diskettenwechsel stimmen deshalb diese Informationen manchmal nicht mehr mit der tatsächlichen Belegung überein. Bei jedem Zugriff auf das Inhaltsverzeichnis berechnet deshalb das BDOS für jeden Record eine Prüfsumme. Ist diese mit der nach dem Start berechneten nicht identisch, so wird die Diskette als nicht beschreibbar gekennzeichnet. Damit werden alle Dateien vor versehentlichem Zerstören geschützt.

»CSV« enthält die Adresse des Speichers, der die Prüfsumme aufnimmt.

Für jeden Block im Datenbereich einer Diskette ist in der Belegungstabelle ein Bit reserviert. Hat dieses Bit den Wert 1, so gilt der Block als belegt. In dem Wort »ALV« steht die Startadresse dieser Tabelle.

Die entsprechenden Routinen des Schneider-Controllers sehen relativ wenig Speicherplatz für die Tabellen »CSV« und »ALV« vor. Falls Sie ein anderes Format wählen, müssen Sie den Tabellen in der Regel einen anderen, genügend großen, Speicherbereich zuteilen.

Der 14 Byte lange »Disketten-Parameter-Block« enthält vom Computertyp unabhängige Informationen über das Format der Disketten. Den Aufbau zeigt Bild 4.

Der 2 Byte lange Wert »SPT« gibt die Anzahl der Records pro Spur an. Der Wert 24 hex (=36 dez) entspricht dann 4,5 (=36*128 Byte) KByte pro Spur.

»BSH« ist die (verschlüsselte) Blockgröße der Diskette. Die Zahl $2^{(BSH)}$ gibt die Anzahl der Records an, die in einem Block stehen. Bei einer Blockgröße von 1 KByte ergibt sich der Wert 3 (1 KByte = 1024 Byte = $128*(2^3)$. Für »BSH« ist 7 als maximaler Wert erlaubt. Die maximale erlaubte Blocklänge beträgt also 16 (=$128*2^7$) KByte.

Der Wert von »BLM« hängt unmittelbar von »BSH« ab. Er errechnet sich nach der Formel BLM=$(2^{(BSH)})-1$. In diesem Byte sind also so viele Bits gesetzt, wie der Wert des BSH-Bytes angibt. Für eine Blocklänge von 1 KByte also 3 Bit und für eine Blocklänge von 16 KByte 7 Bit.

| Bezeichnung | SPT | BSH | BLM | EXM | DSM | DRM | AL0 | AL1 | CKS | OFF |
|---|---|---|---|---|---|---|---|---|---|---|
| Byte | 00-01 | 02 | 03 | 04 | 05-06 | 07-08 | 09 | 0A | 0B-0C | 0D-0E |

**Bild 4. Der Disketten-Parameter-Block ist lebensnotwendig**

»EXM« gibt die um 1 verminderte Zahl der logischen Extents an, die in einem physikalischen Extent enthalten sind. Hat eine Diskette eine Größe von höchstens 256 Blöcken, so können in einem physikalischen Extent maximal 16 Blöcke reserviert werden. Bei einer Blocklänge von 4 KByte sind es 64 (=4*16) KByte und damit 4 logische Extents. Im »EXM« steht dann der Wert 3 (=4–1). Enthält eine Diskette mehr als 256 Blöcke, so können nur noch 8 Blöcke pro Eintrag oder 32 (=4*8) KByte reserviert werden. Das sind dann nur noch zwei logische Extents. In »EXM« steht in diesem Fall der Wert 1 (=2–1). Den genauen Zusammenhang finden Sie in Tabelle 1.

»DSM« bezeichnet je nach Interpretation den um 1 verminderten Wert der Blockanzahl oder die Nummer des letzten Blocks auf der Diskette. Dieser Wert darf problemlos vermindert werden, einige Blöcke auf der Diskette bleiben dann aber ungenutzt. Eine Erhöhung ist jedoch nur erlaubt, wenn auf der Diskette zusätzliche Spuren formatiert wurden. Das BDOS versucht sonst bei einer 3-Zoll-Diskette vergeblich, beispielsweise auf die nicht vorhandene 41. Spur zuzugreifen. Eine Diskette im Systemformat von Schneider enthält bekanntlich 38 Datenspuren zu je 4,5 KByte, also 170 KByte, beziehungsweise Blöcke. Der höchste erlaubte DSM-Wert beträgt damit 169.

In »DRM« steht die um 1 verminderte Zahl der Einträge im Inhaltsverzeichnis, die auf der Diskette vermerkt werden dürfen. Bei einer Vortex unter VDOS 1.0 sind für das Inhaltsverzeichnis 4 KByte reserviert, obwohl nur 2 KByte benutzt werden. Durch Ändern dieses Wertes auf 127 gewinnen Sie – ohne die Disketten neu formatieren zu müssen – 64 zusätzliche Einträge. Bei VDOS 2.0 und folgenden Versionen ist das allerdings schon von Hause aus eingebaut.

Die Bytes »AL0« und »AL1« geben die Anzahl der Blöcke an, die für das Inhaltsverzeichnis reserviert sind. Jedes gesetzte Bit bezeichnet einen Block. Das Bit 7 von »AL0« entspricht dem Block 0 der Diskette. Bit 0 analog Block 7. Bit 7 von AL1 entspricht Block 8 und Bit 0 Block 15. Das Inhaltsverzeichnis darf somit eine Länge bis zu 16 Blöcke annehmen.

Das Wort »CKS« gibt die Größe der CSV-Tabelle des Disketten-Parameter-Headers an. Bei normalen Diskettenstationen stellte sich der Wert (DRM+1)/4 als sinnvoll heraus. Wird der Speicherplatz knapp, ist aber auch ein niedriger Wert zulässig. Bei Festplattenlaufwerken oder RAM-Disks muß man hier den Wert 0 eintragen.

»OFF« gibt die Zahl der Systemspuren auf einer Diskette an. Dieser Wert ist frei wählbar. Üblich ist der Wert 2.

Wie schon weiter oben erwähnt, können Sie bei den Schneider-Computern außer den Standard-Diskettenparametern auch noch einige andere verändern. Der zusätzliche Disketten-Parameterblock ist 10 Byte lang und steht unmittelbar hinter dem Standard-Disketten-Parameterblock. Das 16. Byte des Standard-Blocks entspricht also dem ersten Byte des erweiterten Blocks. Dessen Belegung finden Sie in Bild 5. Sowohl beim Schneider- als auch beim Vortex-Controller, mit oder ohne Speichererweiterung, sind die

| Blockgröße (Byte) | BSH-Wert | BLM-Wert | EXM-Wert | |
|---|---|---|---|---|
| | | | Gesamtblockanzahl < = 256 | Gesamtblockanzahl > 256 |
| 1024 | 3 | 7 | 0 | (verboten) |
| 2048 | 4 | 15 | 1 | 0 |
| 4096 | 5 | 31 | 3 | 1 |
| 8192 | 6 | 63 | 7 | 3 |
| 16384 | 7 | 127 | 15 | 7 |

**Tabelle 1. Die Werte für die Codierung der Blockgröße**

Parameterblöcke fast identisch. Die Unterschiede sind im folgenden extra vermerkt.

Byte 15 enthält die physikalische Nummer des ersten Sektors auf der Spur. Im Systemformat von Schneider steht hier der Wert 41 hex, im Datenformat C1 hex. Das IBM- und das Vortex-Format benutzen den Wert 01 hex.

Byte 16 enthält die Zahl der Sektoren pro Spur – normalerweise 9, nur im IBM-Format 8.

Byte 17 bestimmt die Zeit, in der nach dem Lesen beziehungsweise Schreiben eines Sektors Signale von der Diskette nicht beachtet werden.

Byte 18 steuert den Abstand zweier Sektoren einer Spur beim Formatieren. Er muß so gewählt werden, daß die Sektoren möglichst gleichmäßig auf der Spur verteilt werden. Die genaue Berechnung der beiden eng miteinander korrespondierenden Bytes 17 und 18 ist sehr kompliziert. Experimente helfen schneller weiter, um festzustellen, bei welchen Werten sich Sektoren fehlerfrei schreiben und auch wieder lesen lassen.

Byte 19 enthält den Wert, mit dem die Sektoren beim Formatieren gefüllt werden. In der Regel ist das E5 hex.

Byte 20 enthält den codierten Wert der Zahl der Bytes pro Sektor. Die Sektorlänge berechnet sich dabei mit der Formel $128 * 2^{(Byte\ 20)}$. Für eine Sektorlänge von 256 Byte muß also der Wert 1 eingetragen werden, bei 512 Byte der Wert 2 und bei 1024 Byte der Wert 3.

Leider ist der Sektorpuffer bei Schneider nur 512 Byte lang, so daß Formate mit längeren Sektoren unter CP/M mit 44 KByte TPA nicht problemlos installiert werden können. Anders ist es bei dem CP/M mit Speichererweiterung. In dem 62-KByte-CP/M steht das BIOS im RAM. Mit einem Suchbefehl (beispielsweise »Q« beim DDT) finden Sie alle Stellen im BIOS, an denen die Anfangsadresse des Sektorpuffers benötigt wird. Falls Sie alle Werte korrigieren, dürfen Sie den Puffer in einen anderen Speicherbereich verlegen. Genügend freien Platz zwischen BDOS und BIOS schaffen Sie sich mit »MOVCPM xxx *« und »SYSGEN«.

In Byte 21 steht, wie viele Records in einem Sektor stehen, bei 512 Byte pro Sektor beispielsweise der Wert 4.

Byte 22 dient zum Speichern derjenigen Zylindernummer, über der der Schreib-/Lesekopf gerade steht. Der Wert ändert sich somit laufend.

Byte 23 justiert bei Schneider das Laufwerk unmittelbar vor dem nächsten Diskettenzugriff neu, sofern hier ein Wert ungleich 0 steht. Nach dem Justieren wird es automatisch auf 0 gesetzt. Bei Vortex steht hier die maximale Zylinderzahl des Laufwerks. Bei einem Laufwerk mit 40 Spuren findet sich hier der Wert 27 hex und bei einem Laufwerk mit 80 Spuren 4F hex.

| Byte | Bezeichnung |
|---|---|
| 0F | erste Sektornummer auf der Spur |
| 10 | Anzahl der Sektoren pro Spur |
| 11 | Pause nach Schreiben/Lesen |
| 12 | Abstand zwischen den Sektoren |
| 13 | Füllbyte beim Formatieren |
| 14 | Codierung für Anzahl der Bytes/Sektor |
| 15 | Anzahl der Records pro Sektor |
| 16 | aktueller Zylinder |
| 17 | Schneider: 0 heißt Laufwerk justieren<br>Vortex: maximale Zylindernummer |
| 18 | Schneider: 0 heißt automatische Formaterkennung aktiv<br>Vortex: Flagbyte |

**Bild 5. Mehr Kraft durch den erweiterten Disketten-Parameter-Block**

Auch Byte 24 kommt bei Schneider und Vortex eine unterschiedliche Bedeutung zu. Beim Aufruf der BIOS-Routine zur Laufwerkswahl hat Bit 0 des Registers E immer dann den Wert 0, wenn das Laufwerk seit dem letzten Warmstart zum ersten Mal aufgerufen wird. In diesem Fall stellt die Routine im ROM des Disketten-Controllers von Schneider automatisch das Format der eingelegten Diskette fest. Vor der Rückgabe der Programmkontrolle an das BDOS werden die Daten dieses (in diesem Moment für den Computer neuen) Formats in die Parametertabelle eingetragen. Damit gehen auch von Hand geänderte Daten wieder verloren, da der Computer ja das alte Format restauriert. Wünschenswert hingegen ist, daß ein einmal eingestelltes Format so lange erhalten bleibt, bis man es ausdrücklich wieder ändert. Und es gibt tatsächlich einen Trick, der das bewerkstelligt. Die automatische Formaterkennung wird nämlich nur ausgeführt, wenn in Byte 24 der Wert 0 steht. Ein Eintrag eines anderen Wertes schaltet die Routine zur Formaterkennung ab.

## Experimentierfreude mit fremden Formaten

Von Hand eingegebene Werte bleiben damit so lange erhalten, bis sie ausdrücklich wieder geändert werden. Bei Experimenten mit fremden Diskettenformaten ist es damit immer sinnvoll, als erstes dieses Byte auf den Wert 255 zu setzen. In der Kombination Vortex-Speichererweiterung und Schneider-Controller funktioniert das Abschalten der automatischen Formaterkennung leider nicht.

Das VDOS von Vortex macht prinzipiell keine automatische Formaterkennung. Damit ist Byte 24 aber nicht überflüssig, sondern es stellt das universelle Flagbyte des zugehörigen Laufwerks dar. Jedes Bit hat dabei eine eigene Bedeutung.

Bit 0 hat den Wert 1, wenn ein doppelseitiges Diskettenformat eingestellt ist, und den Wert 0 bei einem einseitigen Diskettenformat. Nur durch Ändern dieses einen Bits läßt sich also der zweite Schreib-/Lesekopf »abschalten«. Unsere Übersetzungsroutine von der Spur- zur Zylinder- und Kopfnummer beachtet dieses Bit ebenfalls.

Bit 1 hat nur bei Laufwerk A eine Bedeutung. Wenn es den Wert 0 hat, wird beim nächsten Warmstart eine Meldung über die aktuelle Betriebssystemversion ausgegeben. Danach wird das Bit automatisch auf 1 gesetzt.

Bit 2 hat ebenfalls nur für Laufwerk A eine Bedeutung – allerdings nur unter VDOS 1.0. Wenn es den Wert 0 annimmt, werden alle ausführlichen BIOS-Meldungen unterdrückt.

Bit 3 hat beim Laufwerk den Wert 1, solange die Motoren eingeschaltet sind. Durch ständige Abfragen können Sie so beispielsweise eine Warteschleife realisieren, die erst dann zum Wechsel der Diskette auffordert, wenn die Motoren stehengeblieben sind. Für Laufwerk B hat dieses Bit keine Bedeutung.

Bit 4 muß den Wert 1 haben, wenn beim nächsten Schreib-/Lesezugriff auf die entsprechende Diskettenstation ein »Multi-Sektor-Transfer« ausgeführt werden soll. In diesem Fall werden mehrere Sektoren auf einmal übertragen. Nach solch einem Zugriff wird das Bit automatisch wieder auf 0 gesetzt. Diese Wahlmöglichkeit kann nur beim direkten Aufruf der entsprechenden XBIOS-Routinen – sowie beim Warmstart – eingesetzt werden.

Wenn Bit 5 in bezug auf Laufwerk A den Wert 0 hat, wird beim nächsten Diskettenzugriff das Diskettenlaufwerk an der Geräteadresse 0 angesprochen. Besitzt dieses Bit hingegen den Wert 1, so wird das Laufwerk an der Geräteadresse 3 angesprochen. Im Disketten-

Parameter-Block von Laufwerk B entscheidet dieses Bit hingegen zwischen den Laufwerken an der Geräteadresse 1 und 3. Eine Änderung dieser Bits (zusammen mit einer Neueinstellung der übrigen Diskettenparameter) entspricht damit den Befehlen »S0« beziehungsweise »S2«.

Bit 6 muß den Wert 0 haben, wenn das zugehörige Laufwerk vor dem nächsten Zugriff justiert werden soll. Danach wird Bit 6 automatisch auf den Wert 1 zurückgesetzt.

Es ist nun sicher nicht jedermanns Sache, mit Debuggern nach einzelnen Bytes im Speicher Ausschau zu halten und anhand der gefundenen Werte die Lage weiterer interessanter Bytes zu bestimmen. Tippen Sie lieber das Turbo-Pascal-Programm aus Listing 2 ab. Es gibt ohne Probleme alle für weitere Experimente interessanten Daten aus.

Hinter der Kennung »dpb:« werden die 15 Byte des Disketten-Parameter-Blocks angezeigt. Hinter »dpx« stehen die 10 Byte des erweiterten Parameterblocks. Beide werden sowohl für Laufwerk A als auch für Laufwerk B auf dem Bildschirm angezeigt. Beachten Sie dabei aber, daß 16-Bit-Werte (ein Doppelbyte) nicht zusammengefaßt, sondern als zwei einzelne Byte ausgegeben werden. Dabei stehen die zwei niederwertigen Stellen im ersten Byte des Paares und die höherwertigen Stellen im zweiten Byte des Paares. Bei den 15 »dpb:«-Byte werden übrigens die Daten berechnet, die auch »STAT DSK:« anzeigt. Mit »STAT« sind sie allerdings so aufbereitet, daß auch der unbedarfte Leser etwas damit anfangen kann. »GETDISK« stellt die Werte hingegen so dar, daß sie in den folgenden Experimenten ohne Umrechnung weiterbenutzt werden können.

»dphstart«, »freistart«, »csvab« und »alvab« bezeichnen die Startadressen von Tabellen und freien Speicherbereichen, die später ebenfalls noch benötigt werden.

Die einzige Prozedur des Programmes installiert eines der in einer Tabelle vorbereiteten Formate. Diese Tabelle läßt sich ohne großen Aufwand um eigene Formate verlängern oder verkürzen. Nur die Konstante »max« ist entsprechend anzupassen. Natürlich können auch vorhandene Formate abgeändert werden. Experimente schaden nichts. Das Schlimmste, was passieren kann, ist, daß auf Ihrer (vorher hoffentlich leeren) Experimentierdiskette unsinnige Daten stehen.

Das Hauptprogramm realisiert eine einfache Menüsteuerung, so daß Sie mit einem Tastendruck eines der vordefinierten Formate installieren können. Danach wird das Programm beendet, aber das entsprechende Format bleibt erhalten. Die Diskette im neuen Format behandeln Sie genauso wie jede andere normale Diskette. Insbesondere können Sie mit »DIR« das Inhaltsverzeichnis anzeigen und mit »PIP« Dateien kopieren. Weil nach dem Ende eines Programms im Laufwerk A eine Systemdiskette liegen muß, ist das Installieren eines neuen Formates in dieser Anwendung nur auf Laufwerk B sinnvoll.

## Was tun mit einem Laufwerk?

Falls Sie nur ein Diskettenlaufwerk besitzen: keine Angst. Sie können die Prozedur in einem eigenen Programm trotzdem einsetzen. Dabei gehen Sie folgendermaßen vor: Im ersten Schritt fordern Sie den Anwender auf, eine Diskette im neuen Format ins Laufwerk A einzulegen. Dann installieren Sie auf dem Laufwerk A dieses Format und rufen danach mit Hilfe der Pascal-Prozedur »bdos(13)« einen Reset der Diskettenstation auf. Im zweiten Schritt lesen Sie wie gewöhnlich mit »reset(datei)...« eine Datei von der Diskette im neuen Format in den Speicher. Im dritten Schritt fordern Sie wieder den Anwender auf, eine Diskette im alten (Schneider-)Format einzulegen, installieren das Originalformat und rufen erneut »bdos(13)« auf. Jetzt können Sie die Daten im Speicher auf die Schneider-Diskette schreiben und das Programm beenden. Falls eine Datei nicht in einem Stück in den Speicher paßt, wiederholen Sie die Schritte eins bis drei mehrmals und kopieren jeweils einen anderen Ausschnitt.

Benutzen Sie ein eigenes CP/M-Format, reichen in der Regel die Tabellen CSV und ALV nicht aus. Diese müssen Sie deshalb in einen genügend großen freien Speicherbereich verschieben – und der liegt natürlich bei jeder Betriebssystemversion an einem anderen Platz. Deshalb kann auch nur ein Sonderformat zur gleichen Zeit aktiv sein. Zwei Formate würden sich nämlich den Platz für die zusätzlichen Tabellen streitig machen. Umgekehrt müssen beim Installieren des Originalformates die CSV- und ALV-Tabellen wieder an der alten Stelle restauriert werden. Im Abschnitt mit den Konstanten müssen Sie deshalb diejenigen Werte einsetzen, die Sie mit »GETDISK« erhalten haben.

Ebenfalls muß das letzte Byte der erweiterten Disketten-Parameter-Tabelle angepaßt werden. Bei einem Vortex-Controller haben Sie vor allem auf die höherwertige Stelle des Byte 24 dieses Bereichs zu achten. Steht hier der Wert 2, wird für dieses Format nicht das Standardlaufwerk benutzt, sondern ein Zusatzlaufwerk. Ungerade Werte bezeichnen immer ein doppelseitiges Format und gerade Werte immer ein einseitiges.

Beim Schneider-Controller muß dieses Byte in allen Formaten den Wert FF hex annehmen. Nur so bleiben neue Formate auch nach einem Warmstart erhalten. Eine Ausnahme bildet das erste Format. Dieses muß vollständig einem der drei Originalformate entsprechen. Das letzte Byte des erweiterten Disketten-Parameter-Blocks muß deshalb wieder den Wert 0 annehmen. Doppelseitige Formate streichen Sie deshalb bei diesem Betriebssystem unbedingt ganz aus der Tabelle.

Zwei bemerkenswerte Formate sind allerdings in der Tabelle enthalten. Ein 3-Zoll-Laufwerk hat nicht nur 40 Spuren, sondern volle 43. Auf eine Spur passen außerdem nicht nur 9 Sektoren, sondern 10. Systemspuren sind bei so besonderen Formaten auch überflüssig. Eine gewöhnliche 3-Zoll-Diskette nimmt so bis zu 215 KByte Daten auf. Das Besondere ist, daß die zusätzlichen 44 KByte nicht mit Tricks zu erreichen, sondern richtig ins Betriebssystem integriert sind. Ein ganz ähnlicher Effekt läßt sich auch bei doppelseitigen Laufwerken mit 80 Spuren erreichen. Bis zu 82 Spuren kann man ansprechen, so daß Sie aus einer 5¼-Zoll-Diskette bis zu 820 KByte Daten hervorlocken können. Durch die Blocklänge von 2 KByte wird nebenbei der Datenbereich bedeutend besser ausgenutzt. Das Inhaltsverzeichnis wurde gleichzeitig auf 6 KByte – und damit 192 Einträge – vergrößert.

## Anregungen

Alle erwähnten Parametertabellen stehen auch unter Basic zur Verfügung, und zwar im Bereich zwischen A700 und AC00 hex. Hier können sie in derselben Weise verändert werden. Sobald Sie die Werte einmal bestimmt haben, haben Sie auch unter Basic mit nur wenigen POKEs die Möglichkeit, das Diskettenformat dauerhaft zu verändern. Mit nur einem Laufwerk kommen Sie zusätzlich zu dem Vorteil, keine Systemspuren für die Informationen zu gebrauchen.

Bisher haben wir nur Formate ohne (logischen) Sektorversatz bearbeitet. Wenn Sie eine BIOS-Routine »SECTRAN« einbauen, die die XLT-Tabelle beachtet, ändern Sie das ohne allzu großen Aufwand.

(Helmut Tischer/hg)

# Routine mit Routinen

**Unter CP/M erleichtern BDOS-, BIOS- und auch Firmware-Routinen die tägliche Arbeit. Doch auch die 13 speziellen System-Routinen für die Diskettenlaufwerke muß man für effektive Programme kennen.**

Die Controller von Schneider und von Vortex kennen 13 spezielle Routinen, die standardmäßig unter CP/M nicht vorhanden sind. Mit ihnen kann man den Datentransfer zwischen Computer und Massenspeicher völlig neu organisieren. Die Routinen sind trotz unterschiedlicher Hersteller bei beiden Betriebssystemen nach außen hin nahezu identisch. Solange im folgenden nichts anderes vermerkt ist, gelten die Anmerkungen für beide Controller.

Bei einem 44 KByte großen CP/M 2.2 liegen die 13 Startadressen für die Routinen im Bereich von BE80 bis BEA6 hex. Das 62 KByte große CP/M (mit Vortex-Speichererweiterung) benutzt für die ersten 11 Adressen die Speicherstellen zwischen F439 und F459 hex. An den Adressen des »kleinen« CP/M stehen »Umleitungen« auf die neuen Werte. So passiert nichts Unangenehmes, wenn ein Programm auf die andere Version von CP/M übernommen wird – sofern der Speicherplatz ausreicht. Die Routinen an den Adressen BEA1 und BEA4 hex sind hingegen mit Speichererweiterung nicht mehr vorhanden. Dafür gibt es zwei neue Routinen und zwar im Speicher bei F433 und F436 hex. Eine kurze Erklärung der einzelnen Routinen (einschließlich Aufruf mit Übergabeparametern und Ergebnis) gibt Tabelle 1. Einige komplizierte Routinen verdienen jedoch noch eine besondere Erklärung.

Nach Fehlern beim Zugriff auf die Diskette erscheint beim Vortex-Controller die Meldung

```
Diskette fehlt. Nochmal versuchen? (J/N)
```

Diese Meldung (bei Schneider erscheint sie auf englisch) stammt nicht aus dem BIOS, sondern wird von der System-Routine an Adresse BE80 hex erzeugt und kann hier auch verhindert werden.

Verschiedene Laufwerkstypen benötigen unterschiedlich viel Zeit, um eine neue Spur aufzusuchen. Mit der Routine von BE83 hex teilen Sie dem BIOS alle wichtigen Laufwerksdaten mit.

An BE86 hex steht die Routine zum Einstellen der Standard-Diskettenformate. Bei Schneider wird im Register E die Nummer des Laufwerks, das ein neues Format erhält, angegeben. Im Register A steht dazu das Format, das eingesetzt werden soll. Mit a=40 hex wird das System-, mit a=C0 hex das Daten- und mit a=00 hex das IBM-Format eingestellt. Die entsprechende Routine von Vortex fordert im Register a den Wert 1, wenn das Laufwerk A durch ein externes Zusatzlaufwerk (beispielsweise eine 3-Zoll-Station) ersetzt werden soll. Mit einer 2 in diesem Register wird Laufwerk B ersetzt. Der Wert 0 steht für die beiden Standard-Laufwerke. Die Formatierung ist einfacher. Eine externe Station wird automatisch mit dem Systemformat von Schneider versehen, das Vortex-Laufwerk natürlich mit diesem Format. Das Programm »S2« auf der System-Diskette von Vortex braucht somit nur 5 Byte lang sein.

```
MVI A,2
JMP BE86
```

Der Vortex-Controller kennt noch zwei weitere Anweisungen. Wird im Register A der Wert FF hex übergeben, so liefert das Registerpaar HL die

| Adresse CP/M 44 KByte | 62 KByte | Funktions-Bezeichnung | Aufruf | Ergebnis |
|---|---|---|---|---|
| – | F433 | Druckerspooler ein/aus | – | Z:0=jetzt aus<br>Z:1=jetzt ein |
| – | F436 | RAM-Disk formatieren | – | – |
| BE80 | F439 | BIOS-Meldungen sperren | A:0=erlauben<br>A:255=sperren | – |
| BE83 | F43C | Zeitkonstanten festlegen | HL:Zeittabelle | – |
| BE86 | F43F | Schneider: Diskettenparameter wählen<br><br><br><br>Vortex: welche Laufwerke aktiv?<br><br><br>nur VDOS 2.0, 62 KByte: Speed on/aus<br><br>Adresse des Flagbytes holen | A:0=IBM<br>A:64=System<br>A:192=Daten<br>E:Laufwerk<br>A:0=intern<br>A:1=A extern<br>A:2=B extern<br>A:254<br><br>A:255<br>E:Laufwerk | –<br><br><br><br>–<br><br><br>A:0=jetzt aus<br>A:255=jetzt ein<br>HL:Flagadresse |
| BE89 | F442 | physikalischen Sektor lesen beide, wenn Bit 4 (Flagbyte)=0<br><br><br>Vortex, wenn Bit 4 (Flagbyte)=1 (Bit 4 wird zurückgesetzt) | E:Laufwerk<br>D:Spur<br>C:Sektor<br>HL:Puffer<br>E:Laufwerk<br>D:Spur<br>C:erster Sektor<br>B:letzter Sektor<br>HL:Puffer | CY:0=Fehler<br>CY:1=in Ordnung<br><br><br>CY:0=Fehler<br>CY:1=in Ordnung |
| BE8C | F445 | physikalischen Sektor schreiben beide, wenn Bit 4 (Flagbyte)=0<br><br><br>Vortex, wenn Bit 4 (Flagbyte)=1 (Bit 4 wird zurückgesetzt) | E:Laufwerk<br>D:Spur<br>C:Sektor<br>HL:Puffer<br>E:Laufwerk<br>D:Spur<br>C:erster Sektor<br>B:letzter Sektor<br>HL:Puffer | CY:0=Fehler<br>CY:1=in Ordnung<br><br><br>CY:0=Fehler<br>CY:1=in Ordnung |
| BE8F | F448 | Spur formatieren | E:Laufwerk<br>D:Spur | CY:0=Fehler<br>CY:1=in Ordnung |
| BE92 | F44B | Kopf positionieren | E:Laufwerk<br>D:Spur | CY:0=Fehler<br>CY:1=in Ordnung |
| BE95 | F44E | Laufwerksstatus feststellen | A:Laufwerk | A:Status<br>CY:0=Fehler<br>CY:1=in Ordnung |
| BE98 | F451 | Wiederholungswert bei Fehler | A:Anzahl | – |
| BE9B | F454 | Externer Zugang zu Firmware-Routinen | (besondere Parameterübergabe) | |
| BE9E | F457 | Schneider: Fast/Slow-Modus wählen<br><br>Vortex: Diskmotoren sofort abschalten | A:0=Slow<br>A:255=Fast<br>– | –<br><br>– |
| BEA1 | – | Schneider: RS232C initialisieren<br>Vortex: Spur-/Zylindernummer | HL:Tabelle<br>A:Spur<br>E:Laufwerk | –<br>A:Zylinder |
| BEA4 | – | Konsolenpuffer füllen (bei Tastendruck Puffer löschen?) | HL:Zeichenkette<br>A:0=löschen<br>A:255=weiter | – |

**Tabelle 1. Die XBIOS-Routinen der Schneider-Computer unter CP/M 2.2**



Adresse des Flagbytes des im Register E angegebenen Laufwerks. Dieses Flagbyte ist identisch mit dem 25. Byte des Diskparameterblocks. Das entspricht damit nicht dem CP/M-Standard. Die Bedeutung der einzelnen Bits erklärt Tabelle 2.

Unter dem VDOS 2.0 des Vortex-Laufwerks wird die beschleunigte Bildschirmausgabe im Modus 2 ein- oder ausgeschaltet, wenn die Routine mit dem Wert FE hex im Register A aufgerufen wird. Diese Anweisung funktioniert allerdings nur bei abgeschalteter Speichererweiterung.

BE89 und BE8C hex lauten die Startadressen zum Lesen beziehungsweise Schreiben eines (meist 512 Byte langen) physikalischen Sektors. Der Befehl wird sofort und ohne Pufferung durch das BIOS ausgeführt. Die Sektoren müssen mit ihren physikalischen Nummern angesprochen werden – bei einer Diskette im Schneider-Systemformat also mit 41 bis 49 hex.

Vortex bietet allerdings noch einen zusätzlichen Befehl. Wenn Bit 4 des Flagbyte des angegebenen Laufwerks gesetzt ist, werden mehrere Sektoren auf einmal übertragen. Register C enthält dann die Kennung des Startsektors und Register B die Nummer des letzten Sektors. Nach Abschluß der Routine wird das Bit automatisch auf Null zurückgesetzt. Diese Methode hat einen sehr schnellen Datentransfer zur Folge.

Mit der Routine von BE8F hex wird jede einzelne Spur einer Diskette formatiert. Dazu werden in einer Tabelle für jeden zu formatierenden Sektor 4 Byte übergeben: die Zylinder-, die Kopf- und die Sektornummer sowie die Sektorgröße. Die Länge der Tabelle wird nicht beim Aufruf mit angegeben, sondern ist in den Diskettenparameterblöcken vermerkt. Eine zu geringe Tabellenlänge führt ohne Fehlermeldung zur fehlerhaften Formatierung. Ist die Tabelle zu lang, bleiben wichtige Sektornummern unberücksichtigt. Ein beliebter Trick ist es, die Sektornummern nicht der Reihe nach anzugeben, sondern folgende (oder eine ähnliche) zu benutzen:

```
1 6 2 7 3 8 4 9 5
```

Das beschleunigt den späteren Zugriff enorm. Beim Vortex-Format muß die Nummer der Spur zuerst in Kopf- und Zylindernummer umgerechnet werden, bevor diese in die Tabelle eingetragen werden darf.

BE92 hex lautet die Startadresse der Routine, die den Schreib-/Lesekopf eines Laufwerks sofort über den Zylinder positioniert. Im Register E wird das Laufwerk und in D die Spur übergeben.

BE95 hex ermittelt den Status eines (in Register A angegebenen) Laufwerks. Damit wird beispielsweise festgestellt, ob die eingelegte Diskette hardwaremäßig schreibgeschützt oder ob überhaupt eine Diskette vorhanden ist.

Die Zahl der Leseversuche bei Übertragungsfehlern legt die Routine an der Startadresse BE98 hex fest.

Normalerweise müssen Sie beim Aufruf von Firmware-Routinen unter CP/M genauestens darauf achten, daß alle – teilweise recht unscheinbare – Nebenbedingungen erfüllt sind. Sonst ist der Computer nur noch durch Aus- und wieder Einschalten zur normalen Arbeit zu bewegen. Die Routine an der Adresse BE9B hex entbindet Sie von dieser Sorge. Der Aufruf ist allerdings etwas ungewöhnlich. Die Register werden wie beim direkten Aufruf von Maschinencode-Routinen mit den Daten geladen. Danach wird der Programmteil immer mit »CALL BE9BH« aufgerufen. Die Adresse der eigentlich gewünschten Firmware-Routine steht in den unmittelbar folgenden Bytes. Zum Ziehen einer diagonalen Linie brauchen Sie beispielsweise folgendes Programm:

```
LXI  D,0000H
LXI  H,0000H
CALL BE9BH
DW   BBC0H
```

| Bit | Wert | Bedeutung |
|---|---|---|
| **Laufwerk A und B:** | | |
| 0 | 0 | bei einseitigem Diskettenformat |
| | 1 | bei doppelseitigem Diskettenformat |
| 4 | 0 | als nächstes nur 1 Sektor Read/Write |
| | 1 | das nächstemal Multi-Sektor-Transfer |
| 6 | 0 | vor nächstem Zugriff Laufwerk justieren |
| | 1 | vor nächstem Zugriff Laufwerk nicht justieren |
| **nur Laufwerk A:** | | |
| 1 | 0 | beim nächsten Warmstart Version anzeigen |
| | 1 | BIOS-Version nicht anzeigen |
| 2 | | (nur bei VDOS 1.0, sonst nicht verwendet) |
| | 0 | BIOS-Fehlermeldungen in Klartext angezeigt |
| | 1 | keine Klartext-Fehlermeldungen |
| 3 | 0 | wenn Laufwerksmotor ausgeschaltet |
| | 1 | wenn Laufwerksmotor eingeschaltet |
| 5 | 0 | Laufwerk auf Geräteadresse 0 verwenden |
| | 1 | Laufwerk auf Geräteadresse 3 verwenden |
| | 7 | (nicht verwendet) |
| **nur Laufwerk B:** | | |
| 1 | | (nicht verwendet) |
| 2 | | (nicht verwendet) |
| 3 | | (nicht verwendet) |
| 5 | 0 | Laufwerk auf Geräteadresse 1 verwenden |
| | 1 | Laufwerk auf Geräteadresse 3 verwenden |
| 7 | | (nicht verwendet) |

**Tabelle 2. Die Bedeutung des Flagbyte des Vortex-Controllers**

| Byte | Größe | Defaultwert | Bedeutung |
|---|---|---|---|
| 0/1 | 16 Bit | 0032 hex | Hochlaufzeit des Diskettenmotors (Einheit 1/50 Sekunde) |
| 2/3 | 16 Bit | | Nachlaufzeit des Diskettenmotors (Einheit 1/50 Sekunde) |
| | | 00FA hex | Schneider: 5 Sekunden |
| | | 0096 hex | Vortex: 3 Sekunden |
| 4 | 8 Bit | AF hex | Wartezeit nach dem Sektorschreiben (Einheit: 1/100 000 Sekunde) |
| 5 | 8 Bit | 1E hex | Grundwartezeit bei Spurwechsel (Einheit: 1/1000 Sekunde) |
| 6 | 8 Bit | | Zusatzwartezeit pro zu wechselnder Spur (Einheit: 1/1000 Sekunde), intern wird immer auf gerade Werte gerundet. |
| | | 0C hex | Schneider: Bits 0–4, Bits 5–7 immer 0 |
| | | C4 hex | Vortex: Bits 0–3 für Laufwerk auf Adressen 0 und 1, Bits 4–7 für Laufwerk auf Adresse 3 |
| 7 | 8 Bit | 01 hex | Bits 0–3: Verzögerung bis zum Abheben des Kopfes nach letztem Zugriff (Einheit: 32/1000 Sekunde) Bits 4–7: immer 0 |
| 8 | 8 Bit | 03 hex | Bits 1–7: Verzögerung nach dem Aufsetzen des Kopfes bis zum ersten Zugriff (Einheit: 1/250 Sekunde) Bit 0 muß immer 1 sein, sonst Systemabsturz |

**Tabelle 3. Die Zeitkonstanten für Diskettenlaufwerke**





```
LXI  D,0279H
LXI  H,018FH
CALL BE9BH
DW   BBF6H
```

Den Routinen an der Adresse BE9E hex kommt bei Vortex und Schneider eine vollkommen unterschiedliche Aufgabe zu. Bei Schneider wird mit ihr zwischen dem Slow- und Fast-Modus hin- und hergeschaltet. Im Slow-Modus dürfen die Register BC' und AF' beliebig verändert werden. Das nützt zum Beispiel Turbo-Pascal aus. Im Fast-Modus ist das nicht gestattet. Dafür laufen aber alle Interrupts und Firmware-Aufrufe bedeutend schneller ab.

Das erste Betriebssystem von Vortex (VDOS 1.0) ließ den CPC immer im Fast-Modus arbeiten. Einige Programme, so zum Beispiel Turbo-Pascal, nehmen das aber übel. Reumütig bekannte sich Vortex zu seinem Fehler. Und mit VDOS 2.0 verweilt der Computer immer im Slow-Modus. Vortex kennt dabei aber bis heute nicht die an dieser Adresse stehende Umschalt-Routine von Schneider. Hier wurde statt dessen eine Routine eingebaut, die den Diskettenmotor augenblicklich abschaltet.

Auch die Routinen an der Adresse BEA1 hex unterscheiden sich bei Vortex und Schneider. Der Schneider-Routine wird hier eine 12 Byte lange Tabelle übergeben. Mit den Daten dieser Tabelle wird die serielle Schnittstelle von Amstrad (nicht zu verwechseln mit der RS232-Schnittstelle von Schneider) initialisiert. Außer dieser Routine enthält das Schneider-BIOS noch einige Ein/Ausgabe-Routinen für diese Schnittstelle, die dabei bereits als ordentliche CP/M-Kanäle implementiert sind.

Die Adresse BEA1 hex steht bei Vortex für eine Routine, die die Umrechnung der Spur- in Zylinder- und Kopfnummer vornimmt. Dabei geht die Unterscheidung zwischen ein- und doppelseitigem Format anhand des Laufwerk-Flagbytes automatisch vor sich.

»CALL BEA4H« entspricht dem Autostart-Kommando. Dieser Routine wird eine Tabelle übergeben, die eine Zeichenkette enthält. Im ersten Byte steht die Länge der Zeichenkette. Danach folgen bis zu 128 Zeichen. Diese Zeichen werden zuerst an die Tastatur »zurückgeschickt«. Eine darauf folgende Konsoleneingabe liefert statt der gedrückten Tasten nach und nach die voreingestellten Zeichen. Erst wenn die Zeichenkette vollständig abgearbeitet ist, wird die Tastatur wie gewohnt behandelt. Auf dem Bildschirm sieht das Resultat dabei so aus, als ob der Computer »seine Tasten selbst drückt«. Mit dem Wert im Register A wählen Sie, ob ein Tastendruck die Auto-Zeichenkette löscht oder nicht.

```
(* CPC-Firmware-Routinen und CPC-XBIOS-Routinen aus Turbo-Pascal aufrufen    *)
(* Version fuer CP/M 2.2, ohne Speichererweiterung                          *)
type register=record f:byte;a:byte;bc:integer;de:integer;hl:integer end;
procedure firmware(var regvar:register;adress:integer);var spvar:integer;begin
inline($2a/adress/$22/*+19/$ed/$73/spvar/$f3/$ed/$7b/regvar/$f1/$c1/$d1/$e1/
$fb/$cd/$9b/$be/$00/$00/$f3/$e5/$d5/$c5/$f5/$ed/$7b/spvar/$fb)end;
procedure xbios(var regvar:register;adress:integer);var spvar:integer;begin
inline($2a/adress/$22/*+19/$ed/$73/spvar/$f3/$ed/$7b/regvar/$f1/$c1/$d1/$e1/
$fb/$cd/$9b/$be/$00/$00/$f3/$e5/$d5/$c5/$f5/$ed/$7b/spvar/$fb)end;
```

**Listing 1. »Firm44« erlaubt Controller-Routinen unter dem normalen CP/M-Format**

```
(* CPC-Firmware-Routinen und CPC-XBIOS-Routinen aus Turbo-Pascal aufrufen   *)
(* Version fuer Vortex-Speichererweiterung, ohne Systemvektor-Verwendung    *)
(* Anpassung an eigene CP/M 2.2 Betriebssystemversion:                      *)
(* Programm 'DDT.COM' starten, danach 'LBB00' tippen.                       *)
(* Es ergeben sich lauter Zeilen '.... CALL yzwx'.                          *)
(* In 'FIRM62.INC'  Zeichenfolge '$XX/$YY/' in '$wx/$yz/' aendern.          *)
(* Vertauschung der Ziffernreihenfolge beachten !                           *)
type register=record f:byte;a:byte;bc:integer;de:integer;hl:integer end;
procedure firmware(var regvar:register;adress:integer);var spvar:integer;begin
inline($ed/$73/spvar/$f3/$ed/$7b/regvar/$f1/$c1/$d1/$e1/$fb/$d9/$21/*+$0e/$e5/
$2a/adress/$23/$23/$23/$e5/$d9/$c3/$XX/$YY/$f3/$e5/$d5/$c5/$f5/$ed/$7b/spvar/
$fb)end;
procedure xbios(var regvar:register;adress:integer);var spvar:integer;begin
inline($ed/$73/spvar/$f3/$ed/$7b/regvar/$f1/$c1/$d1/$e1/$ed/$7b/spvar/$fb/$d9/
$21/*+$0d/$e5/$2a/adress/$11/$b9/$35/$19/$e5/$d9/$c9/$f3/$08/$d9/$2a/regvar/
$11/$08/$00/$19/$f9/$d9/$08/$e5/$d5/$c5/$f5/$ed/$7b/spvar/$fb)end;
```

**Listing 2. Mit »Firm62« kann auch das »aufgeblasene« CP/M Controller-Routinen**

```
program Disketten formatieren;
{$i firmware.inc} (* Firmware-Aufrufe und XBIOS-Aufrufe ermoeglichen *)
const name     = 'Vortex'; (* Name des Formates *)
      maxsec   =   8;      (* Sektoren pro Spur Minus 1 *)
      maxtrack = 159;      (* Anzahl der Spuren Minus 1*)
      sides    =   1;      (* 0 wenn einseitig    formatiert werden soll *)
                           (* 1 wenn doppelseitig formatiert werden soll *)
      formdat: array[0..maxsec,0..3] of byte = ((0,0,1,2),(0,0,6,2),(0,0,2,2),
        (0,0,7,2),(0,0,3,2),(0,0,8,2),(0,0,4,2),(0,0,9,2),(0,0,5,2));
           (* Daten fuer die Sektor-IDs des Zylinders 0, Seite 0 *)
var drive: integer; wahl, taste: char;
procedure formatdisk(drive: integer);
var cpuregister: register; i, track, zylinder, head:integer;
begin
writeln('Format: ',name);
write('Formatieren von Laufwerk ',chr(ord('A')+(drive and 1)),' Zylinder 00');
for track := 0 to maxtrack do
  begin
    if sides = 1
      then begin zylinder := track shr 1; head := track and 1 end
      else begin zylinder := track;       head := 0           end;
    if head = 0 then write(#8#8,zylinder:2);
    for i:= 0 to maxsec do
      begin formdat[i, 0] := zylinder; formdat[i, 1] := head end;
    cpuregister.de := track shl 8 + (drive and 1);
    cpuregister.hl := addr(formdat);
    xbios(cpuregister,$BE8F); (* Spur formatieren *)
    if (cpuregister.f and 1) = 0 then
      begin writeln(' Fehler aufgetreten!'); exit end
  end;
writeln
end;
begin
write('In welchen Laufwerk formatieren? (A/B) ');
repeat read(kbd,wahl);drive:=ord(upcase(wahl))-ord('A')until(drive and 254)=0;
writeln(upcase(wahl));
repeat
  write('Bitte leere Diskette in Laufwerk ',upcase(wahl),' einlegen und ',
    'eine Taste druecken '); repeat until keypressed; writeln;
  formatdisk(drive);
  write('Noch eine Diskette? (J/N) ');
  repeat read(kbd,taste) until (upcase(taste) = 'J') or (upcase(taste) = 'N');
  writeln;
until upcase(taste) = 'N';
write('Bitte eine Systemdiskette in Laufwerk A einlegen und eine Taste ',
  'druecken '); repeat until keypressed; writeln
end.
```

**Listing 3. Formatieren Sie selbst**

Analog zu den Pascal-Standardfunktionen »BDOS« und »BIOS« ist es auch erlaubt, die Firmware- oder Controller-Routinen ohne Kenntnisse von Maschinensprache unter Turbo-Pascal direkt aufzurufen. Die zugehörigen Routinen sind allerdings etwas kompliziert. Sie sehen sie in Listing 1 und 2. Leider ist es unmöglich, sie von der CP/M-Version, mit der man arbeitet, unabhängig zu halten. Es gibt damit also eine Version für die 44 KByte große Version von CP/M 2.2 (Listing 1) und eine für die Version mit Speichererweiterung (Listing 2). Bei der großen Version fallen allerdings die Systemvektoren weg. Sie haben damit unter Turbo-Pascal 57,5 KByte freien Speicherplatz. Im 3. Schneider-Sonderheft von Happy-Computer (Sonderheft 4/1986) finden Sie auf Seite 155 und folgende die ausführliche Beschreibung der Programme. Hier kommt deshalb nur noch einmal das Wichtigste in Kurzform zur Sprache. Zunächst benötigen Sie eine Variable, die ein Abbild des Z80-Registersatzes ist:

```
VAR CPUREGISTER:REGISTER;
```

Diese Variable nimmt die Werte auf, die Sie der Routine übergeben wollen. Falls das Register A den Wert 255 erhalten soll, schreiben Sie:

```
CPUREGISTER.A := 255
```

Den Doppelregistern BC, DE und HL können Sie nichts allein zuweisen, sondern nur zusammen mit dem zweiten Register:

```
CPUREGISTER.BC := B SHL 8 + C
```

Der Prozedur »XBIOS« übergeben Sie das Abbild des Registersatzes und die Adresse der entsprechenden Routine. Zum Abschalten der BIOS-Meldungen beispielsweise:

```
XBIOS(CPUREGISTER,$BE80)
```

Falls die Routine einige Register verändert, sehen Sie die Änderungen im Registersatzbild.

Als Beispiel für die Arbeit mit den XBIOS-Routinen finden Sie in Listing 3 ein Programm, das eine Diskette formatiert. Zu Anfang des Programms bestimmen Sie mit Hilfe von verschiedenen Parametern das Format der Diskette. Die Anzahl der Spuren, Seiten und Sektoren sowie der Inhalt der Sektortabelle stehen hier. Beachten Sie aber, daß die Änderung der Seiten- und der Sektorenzahl nur dann eine Wirkung hat, wenn die neuen Werte zu den in den Parameterblöcken des Laufwerks eingestellten Werten passen.

Bei dem 62 KByte großen CP/M fehlen in der Kopie des BIOS (im RAM-Bereich zwischen F400 und FFFF hex) einige Routinen. In der Systembank sind aber alle originalen XBIOS-Routinen (das sind die Routinen des Controllers) vorhanden. Um dennoch an diese Routinen heranzukommen, gibt es einen Trick. Eine XBIOS-Routine in der Systembank verhält sich nämlich wie eine gewöhnliche Firmware-Routine. Und die Prozedur »FIRMWARE« ist ja dazu da, diese Routinen aufzurufen. Wir benutzen also die Prozedur »FIRMWARE«, übergeben ihr aber nicht die Adresse einer Firmware-Routine, sondern die Adresse einer XBIOS-Routine. Für die beschleunigte Bildschirmausgabe geben wir beispielsweise

```
CPUREGISTER.A:=$FE;
FIRMWARE(CPUREGISTER, $BE86);
```

ein.

Ein weiteres Problem ist die Übergabe von Tabellen an Routinen in der Systembank des Computers. Diese müssen sich im Speicherbereich zwischen 0000 und 7FFF hex befinden. Gewöhnliche Variablen stehen aber am oberen Ende des Speichers. Greifen Sie deshalb zu einer Besonderheit von Turbo-Pascal: der »Const«-Deklaration mit Typangabe. Einerseits können Sie einer derartigen Konstanten während des Programmlaufs wie einer Variable einen neuen Wert zuweisen. Andererseits wird sie aber wie eine Konstante mitten im Programm-Objektcode abgelegt. Und das ist genau das, was wir brauchen. Eine lustige und lehrreiche Anwendung stellt das Programm aus Listing 4 dar.

(Helmut Tischer/hg)

```
program ido; (* Selbstbeschaeftigung *)
{$i firm62.inc} (*hier entweder firm44.inc oder firm 62.inc eintragen *)
const eingabe:string[128]='';
var regvar:register;c:char;
begin repeat
write('Welche Tasten soll ich druecken? '); readln(eingabe);
regvar.hl := addr(eingabe); regvar.a:=0;
firmware(regvar,$bea4);
write('Noch einmal? (J/N) '); read(kbd,c); writeln(c)
until upcase(c) = 'N' end.
```

**Listing 4. »Ido« – Lernen mit Spaß**

# Einzeiler-Wettbewerb

Um festzustellen, zu welchen »Speicherspar«-Leistungen unsere Leser fähig sind, starten wir in dieser Ausgabe einen Einzeiler-Wettbewerb. Alle CPC-Besitzer sind aufgerufen, ein interessantes Basic-Programm zu schreiben. Sie haben bei der Wahl des Themas freie Hand, nur ein einziges Kriterium muß Ihr Programm erfüllen:

Es darf nicht länger als eine Zeile sein. Das Programm muß sich nach Einschalten des Computers an einem Stück direkt über die Tastatur eingeben lassen. Diese Forderung schließt Tricks durch vorausgehende POKE-Befehle aus.

Aber wir setzen auf Ihr Können und Ihre Fantasie. Das Programm kann zum Beispiel ein Utility sein oder einen RSX-Befehl implementieren. Auch Musik- oder Grafikspielereien ohne praktischen Wert sind reizvoll. Wenn Sie es sogar fertigbringen, eine vollständige Textverarbeitung, ein Adventure oder eine Dateiverwaltung in einer Zeile unterzubringen, sind Sie mit Sicherheit Kandidat für einen der folgenden Gewinne:

**1. bis 5. Preis:**

Ein Programm Ihrer Wahl. Es stehen zur Auswahl:

M-Basic, CBasic, Pascal MT+*, Small C*, Wordstar 3.0, dBase II*, Multiplan*, DR DRAW**, DR GRAPH**, oder ein Bücherpaket aus dem Markt&Technik-Verlag im Wert von 200 Mark.

**6. bis 10. Preis:**

Je ein Buchgutschein im Wert von 60 Mark

Zur Teilnahme an unserem Wettbewerb müssen Sie lediglich Ihren Einzeiler mit einer ausführlichen Programmbeschreibung (Definition der Variablen, Eingabe von Parametern etc.) bis zum **1. Februar 1987** (Datum des Poststempels) an

**Redaktion Happy-Computer**
**»CPC-Einzeiler-Wettbewerb«**
**Markt&Technik Verlag AG**
**Hans-Pinsel-Straße 2**
**8013 Haar bei München**

einsenden. Sie können auch mit mehreren Einzeilern am Wettbewerb teilnehmen. Der Rechtsweg ist ausgeschlossen.

Damit sich mit den Gewinnern auch unsere Leser freuen können, werden wir die besten Einzeiler veröffentlichen. Programmierer, deren Programme nicht prämiert wurden, jedoch so interessant sind, daß wir Sie den Lesern nicht vorenthalten möchten, erhalten ein angemessenes Honorar. (ma)

* 128 KByte RAM erforderlich
** nur für CPC 6128

# Computerwissen von A bis Z

Mehr als 160 Seiten mit Informationen über Ihren Schneider CPC liegen vor Ihnen. Ganz klar, daß jede Menge Fachworte in den einzelnen Artikeln auftauchen. Damit jeder Leser eine Chance hat, das Computer-Chinesisch auch richtig zu verstehen, finden Sie hier die wichtigsten Worte erklärt. Weit über 100 Stichpunkte von Abschirmung bis Zylinder werden erläutert.

(hg/ma)

**Abschirmung:** Metallgehäuse oder Drahtgeflecht, das eine Schaltung oder eine Leitung vor der Einwirkung von Störstrahlen (durch Radio, Fernsehen, und so weiter) schützt oder die Abstrahlung von Störimpulsen verhindert.

**Absolutwert:** Vorzeichenloser Wert einer Zahl. So ist der Absolutwert von -5 der gleiche wie von +5, und zwar der Wert 5.

**Adreßbus:** Gruppe von Signalleitungen, die der Mikroprozessor zur Adressierung von Speicher und Peripherie benutzt.

**Algorithmus:** Schema für einen Programmablauf. Jedes Problem, das ein Computer bearbeiten kann, läßt sich schematisch darstellen - und damit als Algorithmus formulieren.

**Amplitude:** Größter Wert, den eine periodische Schwingung annehmen kann. Bei einem Uhrpendel ist die Amplitude der maximale Ausschlag in eine Richtung.

**Animation:** Simulation einer Bewegung durch schnelle Wiedergabe von Einzelbildern.

**ASCII:** (American Standard Code for Information Interchange) Genormter Code für die Zeichendarstellung und -übertragung bei Computern. Der normale ASCII-Zeichensatz arbeitet mit nur 7 Bit (128 verschiedene Symbole), so daß das 8. Bit eines Bytes von verschiedenen Computerherstellern unterschiedlich genutzt wird.

**Assembler:** Maschinenorientierte Programmiersprache. Im Gegensatz zu Hochsprachen ist ein Assemblerprogramm auf einen speziellen Prozessor ausgelegt und damit für den Menschen relativ schwer zu verstehen.

**asynchron:** Gegensatz zu synchron (gleichlaufend). Mit asynchron werden Programmteile bezeichnet, die unabhängig vom gerade bearbeiteten Programm durch ein anderes Ereignis, beispielsweise einer Zeitvorgabe, aufgerufen werden.

**BDOS:** (Basic Disk Operating System) Hardwareunabhängiger Teil des Betriebssystems CP/M. Das BDOS ist bei allen CP/M-Computern gleich.

**Betriebssystem:** Routinen, die zum Betrieb eines Computers und seiner Peripherie unbedingt erforderlich sind. Diese Programme steuern die gesamte Verwaltung von Speicher, Bildausgabe, Datenübertragung etc.. Ohne ein Betriebssystem ist ein Computer nicht arbeitsfähig.

**binär:** Auf einem Zahlensystem basierend, das nur zwei verschiedene Werte kennt. Während das normalerweise benutzte Dezimalsystem 10 verschiedene Ziffern kennt, arbeitet das Binärsystem nur mit den Werten 0 und 1. Die Ziffer 2 benötigt schon zwei Stellen (10) und die 4 sogar drei (100).

**Binärdatei:** Prinzipiell jede Datei, die Daten im Binärcode enthält. Beim Schneider CPC werden als Binärdatei Datensätze bezeichnet, die ausschließlich aus Maschinencode-Daten bestehen. Dabei handelt es sich meist um Maschinencode-Programme, aber auch um gespeicherte Bildschirminhalte.

**BIOS:** (Basic Input Output System) Hardwareabhängiger Teil des Betriebssystems CP/M. Das BIOS enthält die computerspezifischen Ein- und Ausgabe-Routinen eines CP/M-Computers.

**Bit:** (binary Digit) Kleinste Speichereinheit in einem Computer. In ihr kann der Wert 1 (Spannung ein) oder 0 (Spannung aus) stehen. Normalerweise können 8 Bit gemeinsam als Byte angesprochen werden. Daraus ergibt sich der maximale Wert von 256 Zuständen ($=2^8$), der in einer Speicherzelle (=1 Byte=8 Bit) abgelegt werden kann.

**Blockgrafik:** Grafik, die sich aus dem Zeichensatz des Computers zusammensetzt. Bei den meisten neueren Geräten können die Symbole per Programm geändert werden. Damit lassen sich Bilder ähnlich der hochauflösenden Grafik - hierbei wird jeder Punkt auf dem Bildschirm einzeln angesprochen - erzeugen.

**Boolesche Algebra:** Logische Verknüpfung zweier Informationen. Bei dieser Art der Mathematik werden logische Zusammenhänge in Formeln, wie sie in der normalen Algebra benutzt werden, niedergeschrieben.

**Booten:** Laden des Betriebssystems von Diskette.

**BTX:** Bildschirmtext ist ein Kommunikationsmittel, bei dem das Fernsehgerät als Terminal und das Telefon als Datenleitung dient. Der Zweck ist, die Leistungen eines Großcomputers - und damit auch Dienstleistungen verschiedener Firmen - in jeden Haushalt zu bringen.

**Byte:** 8 Bit, die zu einer Informationseinheit zusammengesetzt werden. Ein Byte ist die kleinste Einheit, die man im Speicher eines Schneider-Computers direkt ansprechen kann.

**Centronics:** Amerikanischer Druckerhersteller. Die parallele Schnittstelle, die Centronics in seine Drucker einbaut, hat sich als Norm durchgesetzt. Mit ihr werden 8 Bit parallel übertragen.

**Compiler:** Routine, die ein im Quellcode (für den Menschen verständlich) geschriebenes Programm in eine für den Computer zu bearbeitende Datei umsetzt. Fast alle Computersprachen arbeiten mit einem Compiler.

**Controller:** Baustein oder Baugruppe, die eine Hardware-Einheit kontrolliert (zum Beispiel Disketten-Controller).

**CP/M:** (Control Program for Microcomputers) Vom Computer unabhängiges Betriebssystem für 8-Bit-Geräte. Voraussetzung ist eine CPU vom Typ 8080, 8085 oder Z80. CP/M erlaubte zum ersten Mal, Programme unabhängig vom Computertyp zu schreiben. Heute ist CP/M im professionellen Bereich von MS-DOS verdrängt worden, erlebt bei den Heimcomputern aber eine zweite Blüte.

**CPU:** (Central processing unit, zu deutsch: zentrale Prozessoreinheit). Wird als Abkürzung für Mikroprozessor benutzt, dem Herzen eines Computers.

**Cracker:** Computerbenutzer, der Dateien mit Kopierschutz knackt und sich auf diese Weise Zugang zu geschützten Daten verschafft. Im Heimbereich ist das »Cracken« meist auf Spiele beschränkt.

**Cursor:** Zeichen auf dem Bildschirm, das den Punkt für die nächste Eingabe markiert.

**DATA-Lader:** Basic-Programm, das Werte aus DATA-Zeilen als Maschinencode-Programme direkt im Speicher ablegt.

**Datenbus:** Gruppe von Signalleitungen, die von der Hardware des Computers und der Peripherie zum Austausch von Daten verwendet wird.

**Daten-Format:** Aufzeichnungsformat bei Schneider-Disketten, bei dem kein Platz für Systemspuren freigehalten wird.

**Datex-P:** Postalisches Netz zur Datenfernübertragung. Datex-P ist technisch dem Telefonnetz sehr ähnlich. Die Gebühren sind allerdings bedeutend niedriger.

**DDT:** (Dynamic Debugging Tool) CP/M-Programm zur Fehlersuche.

**dezimal:** Normales Zahlensystem mit 10 verschiedenen Ziffern.

**DFÜ:** Datenfernübertragung. Informations- und Programmaustausch zwischen zwei Computern über das Postnetz.

**Dialog:** Verständigung (»Unterhalten«) zwischen zwei Computern oder Menschen und Computern.

**Digitalvoltmeter:** Spannungsmeßgerät mit Ziffernanzeige.

**Directory:** Verzeichnis aller Daten auf einer Diskette. Bestimmte Spuren auf einem Datenträger sind für dieses Verzeichnis freigehalten. Mit seiner Hilfe findet der Computer die einzelnen Daten.

**DMA:** (Direct memory access) Elektronischer Baustein, der die Speicherverwaltung übernimmt und dadurch den Mikroprozessor entlastet. Da der DMA auf die Übertragung großer Datenmengen spezialisiert ist, macht er das sehr schnell.

**DOS plus:** (Disk Operating System) Betriebssystem für 16-Bit-Computer mit einer CPU von Intel. DOS plus ist die Antwort von Digital Research auf MS-DOS. DOS plus beherrscht den Befehlssatz von CP/M 86 (CP/M-Version für 16-Bit-Computer) und MS-DOS.

# Computerwissen von A bis Z

**Double Density:** Aufzeichnungsformat auf Disketten mit doppelter Datendichte.

**Download:** Übertragung von Daten aus einer Datenbank in den eigenen Computer.

**Editor:** Programm zum Eingeben von Texten und Programmen.

**Elektrolytkondensator:** Spezielle Bauart eines Kondensators, die höhere Kapazitäten bei gleichen Abmessungen erlaubt. Besitzt in der Regel eine vorgeschriebene Polung.

**Escape-Sequenz:** Zeichenfolge, die der Computer an den Drucker sendet, um anzukündigen, daß die folgenden Daten nicht gedruckt werden sollen, sondern dem Einstellen auf bestimmte Arbeitszustände dienen.

**Extension:** Kombination aus drei Buchstaben, die den Dateityp angibt (BAS für Basic-Programme, BIN für Binärfelder und so weiter).

**Fast-Ticker:** Interruptuhr beim Schneider, die alle 1/300-Sekunde berücksichtigt wird.

**Festplatte:** Speichermedium, das aus festen, magnetisierbaren Scheiben (Harddisks) aufgebaut ist. Die geschlossene, staubgeschützte Bauweise erlaubt sehr hohe Speicherdichten (20 bis 60 MByte) und rasante Datenübertragung durch höhere Drehzahlen.

**Firmware:** Fest eingebautes Betriebssystem eines Computers.

**Flag:** Bit oder Byte, das bei Entscheidungen gesetzt wird und in der späteren Ausführung zu unterschiedlichen Bearbeitungsarten führt.

**Floppy:** Anderes Wort für Diskette

**Formatieren:** Vorbereitung der Diskette für die Datenspeicherung. Erst dadurch wird es dem Controller ermöglicht, abgelegte Daten wieder zu finden.

**Frequenz:** Anzahl von Schwingungen eines Signals in einer Sekunde.

**Gate-Array:** Standard-IC mit gitterförmig angeordneten Bauelementen. Die endgültige Verdrahtung der Elemente, und damit die spezielle Funktion des Bausteins, bestimmt der Kunde des IC-Herstellers. Viele Computerhersteller bauen in ihre Geräte solche Bausteine ein, um die Zahl der Einzelelemente zu minimieren – und Hardware-Nachbauten zu erschweren.

**GEM:** (Graphics Environment Manager). Grafische Benutzeroberfläche, bei der einzelne Betriebssystemaufrufe durch Symbole dargestellt werden. Der Sinn von GEM ist die Nachbildung eines elektronischen Schreibtisches, der für den normalen Computerbenutzer leichter zu bedienen ist als direkte Betriebssystem-Befehle.

**globale Variable:** Variable, die im gesamten Programm gültig ist.

**GSX:** (Graphics Extension) Grafische Erweiterung von CP/M plus.

**Hacker:** Computerbesitzer, der über DFÜ in fremde Computersysteme eindringt, Daten liest und teilweise manipuliert.

**Hardcopy:** Ausdruck des Bildschirms auf einen Drucker.

**Hardware:** Feste, greifbare Bestandteile eines Computersystems (Gehäuse, Bauteile, Drähte und so weiter).

**Hashing:** Algorithmus zum Sortieren von Daten. Unser Listing CPC benutzt dieses Verfahren, um bei der Eingabe vertauschte Werte zu erkennen.

**Hertz:** Maßeinheit für Frequenz. Gibt Schwingungen pro Sekunde an.

**hexadezimal:** Zahlensystem, das 16 verschiedene Ziffern benutzt. Da 16 ein Vielfaches von 2 (der Grundzahl des Binärsystems) ist, erlaubt das hexadezimale System eine besonders übersichtliche Darstellung der Speicherinhalte von Computern. Die Zahl 255 wird noch mit zwei Ziffern (FF) geschrieben.

**Hexdump:** Hexadezimale Wiedergabe eines Speicherbereichs.

**Hochsprache:** Programmiersprache, die an die menschliche Darstellungskraft angelehnt ist. Im Gegensatz zu Assemblersprachen sind Hochsprachen »relativ« einfach zu verstehen. Die Umsetzung in eine für den Computer verständliche Sprache ist aber oft sehr kompliziert.

**IBM-Format:** Aufzeichnungsformat bei Schneider-Disketten, das auch von IBM benutzt wird.

**ID-Byte:** Erkennungsbyte (Flag) bei Turbo-Pascal.

**Include-Dateien:** Programmteile, die unter Turbo-Pascal erst beim Compilieren in den Quelltext eingebaut werden.

**Integral:** Mathematische Berechnung einer Differenzialgleichung. Mit Hilfe eines Integrals wird die Fläche unter einer Kurve berechnet.

**Interface:** Bindeglied zwischen Computer und Peripherie.

**Interpreter:** Im Gegensatz zu einem Compiler übersetzt der Interpreter ein Programm schrittweise in für den Computer verständliche Maschinencode-Anweisungen. Dadurch wird der Programmablauf langsam, die Fehlersuche ist aber einfacher. Basic ist eine typische Interpretersprache.

**Interrupt:** Unterbrechung eines Programms in Abhängigkeit von der Zeit und unabhängig vom gerade bearbeiteten Programmteil.

**I/O-Adresse:** Adresse für ein Port (Tor) zur Ein- und Ausgabe von Daten.

**Kaltstart:** Neustart des Computers mit vollständigem Löschen des Speichers.

**kompatibel:** verträglich. Kompatible Computer verarbeiten problemlos die gleiche Software. In der Regel sind Heimcomputer verschiedener Hersteller untereinander nicht kompatibel. Oft auch als Kurzbezeichnung für Computer, die zum IBM-PC kompatibel sind.

**Label:** Marke in einem Programm, die vor dem Start der Software in eine effektive Adresse umgerechnet werden muß. Besonders in der Maschinensprache haben Labels eine große Bedeutung, da das Programm erst zum Laufen an der endgültigen Adresse stehen muß.

**Leiterbahn:** Leitende Verbindung zweier Punkte auf einer Platine.

**Leitungstreiber:** Verstärker für digitale Signale.

**Linker:** Routine, die einzelne Programmteile zu einem zusammenhängenden Code zusammensetzt und zusätzliche Routinen aus anderen Quellen einbindet.

**Listing:** Ausdruck des Programm-Codes.

**lokale Variable:** Variable, die nur in einem bestimmten Programmbereich gültig ist.

**Makroassembler:** Übersetzungsprogramm von Assemblersprache in Maschinencode, den die CPU versteht, und das erlaubt, eigene Befehle zu definieren.

**Mailbox:** Elektronischer Briefkasten für mit dem Computer zu übertragende Informationen.

**Maschinencode:** Anweisungen in einer für die CPU verständlichen Form. Häufig wird Assembler (eine für den Menschen verständliche Form des Maschinencodes) mit Maschinencode verwechselt.

**Maus:** Gerät, das durch Bewegen über eine glatte Oberfläche den Cursor (Pfeil) auf einem Computer-Bildschirm steuert.

**Menü:** Bildschirmgestaltung, die die Arbeit mit einem Programm erleichtert. Bei einem Menü werden Anweisungen durch Drücken einer Taste oder durch Anwahl mit einer Maus oder ähnlichem und nicht durch Eingabe des Befehlswortes erteilt.

**Mikroprozessor:** (CPU) Zentraler Baustein eines Computers, der für den Programmablauf und die Datenverarbeitung zuständig ist. Der Mikroprozessor kommuniziert über Adreß- und Datenbus mit den anderen Einheiten des Computers.

**MS-DOS:** (Microsoft Disk Operating System) Betriebssystem für 16-Bit-Computer mit CPU von Intel.

**Multitasking:** Paralleler Ablauf verschiedener Programme.

**Netzteil:** Schaltung (meistens berührungssicher in ein Gehäuse eingebaut), die Netzspannung in Niederspannung für empfindliche Elektronik (zum Beispiel Computer) umwandelt.

**Netzwerk:** Verbindung mehrerer allein arbeitender Computer. Über das Netzwerk werden Informationen zwischen den Einzelgeräten ausgetauscht.

**Offset:** Differenz zwischen zwei Adressen.

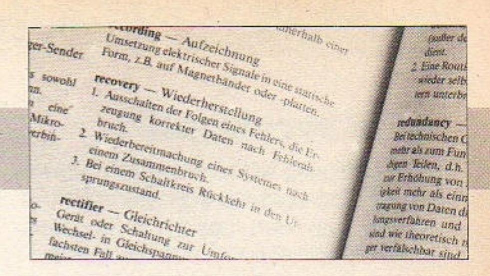
**Oszillograph:** Meßgerät, das Spannungsverläufe auf Papier grafisch darstellt.

**Paritätsbit:** Gibt an, ob die Zahl der gesetzten Bits in einem Byte gerade oder ungerade ist.

**Pascal:** Programmiersprache, die wegen ihrer Struktur besonders zum Erlernen des Programmierens geeignet ist.





**Patch:** Flicken. Patches dienen dem Beheben kleiner Programmfehler und werden direkt in den Computer eingegeben oder auf den Datenträger geschrieben.

**Pi:** Unendliche nichtperiodische Zahl, die das konstante Verhältnis des Kreisumfangs zum Durchmesser angibt (Pi ist näherungsweise 3,141593).

**Pixel:** Bildschirmpunkt

**Portadresse:** Adresse für ein Tor zur Peripherie, über das Daten mit der CPU ausgetauscht werden.

**Potentialabgleich:** Zwei Schaltungen mit getrennter Spannungsversorgung, die miteinander verbunden werden sollen, müssen einen gemeinsamen Bezugspunkt für die Spannungswerte erhalten. Der Bezugspunkt wird in der Regel hergestellt, indem die Masse der ersten Schaltung an die Masse der zweiten angeschlossen wird.

**Potentiometer:** Regelbarer Widerstand.

**Proportionalschrift:** In einem Text, der mit Proportionalschrift gedruckt wurde, sind nicht alle Zeichen gleich breit, sondern nehmen nur so viel Platz ein, wie sie unbedingt benötigen. Dadurch wird zum Beispiel ein »i« schmaler als ein »m«. Auch die Schrift, die Sie gerade lesen, ist Proportionalschrift.

**Puffer:** Speicherbereich, der zum Zwischenlagern von Daten dient. Langsame Peripheriegeräte, beispielsweise ein Drucker, hindern somit nicht mehr den Computer am Weiterarbeiten, da die Daten schnell in den Puffer geschrieben werden und dann unabhängig vom Programm langsam an das Peripheriegerät weitergeleitet werden.

**Pull-Down-Menü:** Menü, das durch Anklicken eines Symbols in der oberen Bildschirmzeile aktiviert wird und seinen Menütext aus der Kopfzeile heraus auf den Bildschirm herunterschreibt.

**Quellcode:** Programmtext, wie er vom Menschen eingegeben wird.

**RAM:** Arbeitsspeicher, dessen Inhalt der Anwender ändern kann. Der Speicherinhalt geht jedoch beim Ausschalten des Gerätes verloren.

**Record:** Die Datenübertragung zwischen Diskette und Computer verläuft unter CP/M immer in Portionen zu 128 Byte. So eine Portion heißt Record.

**Referenzspannung:** Bezugsspannung für elektrische Meß- und Umwandlungsvorgänge.

**Register:** Speicherzellen innerhalb der CPU, die sehr schnell angesprochen werden können und damit für die Arbeit der CPU unverzichtbar sind.

**relokatibel:** Programme, die überall im Speicher liegen und arbeiten können.

**Reset:** Rücksetzen des Computers in einen definierten Ausgangszustand. Alle gerade ausgeführten Aktivitäten werden abgebrochen.

**ROM:** Festwertspeicher, der das Betriebssystem und weitere feste Daten des Computers enthält. Der Inhalt kann vom Anwender nicht überschrieben und nicht gelöscht werden.

**RS232C:** Norm für eine serielle Schnittstelle.

**RSX:** (Resident System Extention) Basic-Erweiterung beim Schneider-Computer, die dauerhaft in das Betriebssystem eingebunden wird. Die Anweisungen sind durch einen »I« gekennzeichnet.

**Runtime-Error:** Fehler, der während des Programmlaufs auftritt.

**Schnittstelle:** Gruppe von Signalleitungen, die für den Anschluß einer Peripherie-Einheit auf eine gemeinsame Buchse (oder Stecker) geführt sind.

**Schrittmotor:** Elektrischer Motor, der sich bei jedem Spannungsimpuls um einen definierten Winkel dreht.

**Scrollen:** Verschieben des gesamten Bildschirms um eine Position (normalerweise nach oben).

**Sektor:** Jede Spur einer Diskette ist in verschiedene Sektoren eingeteilt. Ein Sektorbereich sieht anschaulich wie ein Tortenstück bei einem runden Kuchen aus.

**Single Density:** Einfache Schreibdichte auf einer Diskette (in der Regel im 40-Spur-Betrieb).

**Slow-Ticker:** Zeitgeber des Schneiders, der alle 1/50-Sekunden aufgerufen wird.

**Software:** Programme, die einen Computer erst zu einer leistungsfähigen Maschine machen.

**Spannungsteiler:** Serienschaltung von Widerständen, die eine Gesamtspannung in Einzelspannungen aufteilt.

**Speichererweiterung:** Interne oder externe Computer-Schaltung, die den Festwertspeicher oder Arbeitsspeicher vergrößert.

**Speicheroszilloskop:** Elektronisches Meßgerät, das Spannungsverläufe auf einem Bildschirm grafisch darstellt und den Signalverlauf speichert.

**Steuerzeichen:** Optisch nicht darstellbares Zeichen, das eine Aktion auslöst. So schaltet zum Beispiel <CTRL+P> unter CP/M das Druckerprotokoll ein.

**synchron:** (gleichlaufend) Synchrone Ereignisse werden vom Computer »passend« (und damit meist vom Programm aus) aufgerufen. Gegensatz: asynchron.

**SYS-Datei:** Datei, die vom Betriebssystem für seine Arbeit direkt gebraucht wird.

**Syntax:** Grammatik einer Programmiersprache. In der Syntax steht, wie ein Befehl geschrieben werden muß und mit welchen Parametern oder anderen Anweisungen er kombiniert werden darf.

**System-Format:** Beim Schneider wird das Diskettenformat, das die CP/M-System-Spuren enthält, als System-Format bezeichnet.

**Systemspuren:** Die Spuren einer Diskette, auf denen das Betriebssystem (CP/M oder MS-DOS) steht.

**Taktfrequenz:** Rhythmus, auf den sich zeitlich alle Vorgänge im Computer beziehen.

**Tastaturdecodierung:** Schaltung, die registriert, welche Tasten einer Tastatur gedrückt werden und in ein spezielles Zeichen umsetzt. So kann der Computer erkennen, welche Taste gedrückt wurde.

**Terminal:** Gerät, das der Kommunikation zwischen Computer und Benutzer dient. Beim Schneider sind – wie bei Heimcomputern üblich – alle Terminalfunktionseinheiten (Bildschirm und Tastatur) bereits im Computer integriert.

**Timer:** Zeitgeber

**Tongenerator:** Programmierbare Schaltung oder Baustein zur Erzeugung von Tönen und Geräuschen.

**TPA:** (Transient Program Area) Speicherbereich unter CP/M, der für Programme frei ist.

**Turtle:** Grafikcursor unter Logo (und manchmal auch Pascal).

**User:** Anwender. Im weiteren Sinne Speicherbereiche. Der Schneider unterstützt – wie auch CP/M – 16 verschiedene Datenbereiche. In diesen lassen sich logisch zusammenhängende Programme zusammenfassen.

**Utility:** Hilfsprogramm

**Vektor:** Sprungadresse zu bestimmten Speicherbereichen.

**Vibrato:** Vibrationen eines Tonsignals durch geringfügige Frequenzschwankungen.

**Warmstart:** Start eines Programms nach einer Unterbrechnung, wobei die Daten noch im Speicher vorhanden sind.

**Window:** Der Schneider erlaubt die Einteilung des Bildschirms in acht Bereiche. Diese Windows werden wie kleine Bildschirme behandelt.

**Z80:** Weit verbreiteter Mikroprozessor, der auch in den Schneider CPCs verwendet wird.

**Zeichensatz:** Sammlung aller vom Betriebssystem definierten Zeichen, die der Computer auf dem Bildschirm darstellen kann. Der Zeichensatz unterteilt sich in Sonderzeichen, alphanumerische Zeichen und Grafikzeichen. Die Sonderzeichen liegen im ASCII-Code-Bereich von 0 bis 31 und werden in der Regel mit Hilfe der CTRL-Taste aufgerufen. Die alphanumerischen Zeichen repräsentieren die ASCII-Codes 32 bis 127, und die Grafikzeichen schließen sich bis zur Nummer 255 an. Besitzer des CPC können sämtliche Zeichen durch den SYMBOL-Befehl umdefinieren.

**Zenerdiode:** Diode, die in Sperrichtung in eine Schaltung eingebaut wird und bei Überschreiten einer definierten Spannung durchschaltet. Auf diese Weise lassen sich Überspannungen kurzschließen und Versorgungsspannungen stabilisieren.

**Zylinder:** Bei einer Festplatte haben auf den verschiedenen Scheiben übereinanderliegende Spuren immer die gleiche Kennziffer. So einen gemeinsamen Bereich auf mehreren Scheiben bezeichnet man als Zylinder. Auf einer Diskette sind Zylinder und Spur das gleiche.

## Nachhall

Sehr positiv wurde unser erster Sonderheft-Nachhall aus der letzten Ausgabe von den Lesern aufgenommen. Diesmal bieten wir Ihnen die Verbesserungen und Berichtigungen zu allen anderen Schneider-Sonderheften in dieser Form an. Besonders der SMON-Patch für den Maschinensprache-Monitor und die Hinweise zur Anpassung des Disk-Monitors aus der letzten Ausgabe an alle CPC-Modelle sind wertvolle Informationen für jeden Leser. Doch auch die übrigen Tips&Tricks helfen Ihnen, Programme und Schaltungen aus alten Schneider-Sonderheften optimal einzusetzen.

## SMON stark verbessert

**So passen Sie den Maschinensprache-Monitor an alle CPC-Modelle an.**

Im zweiten Schneider-Sonderheft haben wir ab Seite 58 den Maschinensprache-Monitor SMON veröffentlicht. Für jeden Assembler-Programmierer ist dieses Programm eine große Hilfe, und die begeisterten Leserreaktionen bestärkten uns in unseren Bemühungen, den Monitor zu verbessern und an alle drei CPC-Modelle optimal anzupassen.

So haben wir ein kleines Patch-Programm entwickelt, das den Maschinensprache-Monitor selbständig vom Datenträger lädt, verbessert, nach den Wünschen des Anwenders abändert und anschließend die neue Version automatisch speichert.

Die RSX- und Datei-Befehle funktionieren danach auch auf dem CPC 664 einwandfrei. Die Formatierung der Bildschirmausgabe auf dem CPC 664 und 6128 wird verbessert. Auch der Übersetzungsfehler, der den Z80-Befehl »LD A,(nnnn)« irrtümlicherweise als »LD HL,(nnnn)« interpretiert, ist beseitigt.

Die Bildschirmparameter für SMON können Sie nach den eigenen Wünschen festlegen. Die Wahl zwischen Modus 1 und 2 ist ebenso erlaubt wie das Setzen der persönlich favorisierten Zeichen-, Hintergrund- und Rahmenfarbe; das Ganze immer unter der Voraussetzung, daß das Original-Programm SMON als Datei mit dem Namen »smon.bin« auf dem gleichen Datenträger wie das Patch-Programm vorhanden ist.

Unser Listing zeigt das Patch-Programm und macht deutlich, daß nur wenige Programmzeilen wesentliche Verbesserungen erzielen. Nach dem Programmstart wird der Monitor SMON an seine Startadresse im Speicher ab Adresse 8000 hex geladen. Das Patch-Programm stellt automatisch fest, um welches CPC-Modell es sich handelt und modifiziert den SMON entsprechend. Anschließend liest es die Daten für Bildschirm-Modus und Farben direkt aus dem SMON heraus und zeigt sie an. Der Anwender wählt entweder neue Werte oder bestätigt die alten Daten durch erneute Eingabe.

Sind alle Abfragen beendet, wird die neue SMON-Version gespeichert. Die Wirkung der neuen Einstellungen können Sie sofort mit dem Befehl »CALL &8000« überprüfen.

(Volker Everts/ma)

```
1 REM SMON-PATCH                              [E25E]
2 REM ----------                              [0930]
3 REM (C) 1986 Volker Everts                  [CF1E]
4 REM                                         [2870]
5 MEMORY &7FFF:IF PEEK(&8000)<>49 THEN L
  OAD"SMON",&8000                             [45B6]
10 IF PEEK(6)=128 THEN CPC=1:CPC$="464":
   RESTORE 100                                [D092]
15 IF PEEK(6)=&7B THEN CPC=2:CPC$="664":
   RESTORE 200                                [93AC]
20 IF PEEK(6)=&91 THEN CPC=3:CPC$="6128"
   :RESTORE 300                               [D3EC]
25 PRINT:PRINT"** SMON-PATCH CPC-";CPC$:
   PRINT                                      [D018]
30 READ ED:POKE &81DE,ED 'EDIT-Aufruf         [C142]
35 READ Z:POKE &82C8,Z:POKE &8573,Z:POKE
    &8817,Z:POKE &89B2,Z 'Textpuffer          [53BE]
40 READ Z:POKE &81E2,Z:POKE &89B8,Z:POKE
    &8A01,Z 'Cursorspalte Low                 [29FC]
45 READ Z:POKE &81E3,Z:POKE &89B9,Z:POKE
    &8A02,Z 'Cursorspalte High                [E1A8]
50 READ Z:POKE &8100,Z:READ Z:POKE &8101
   ,Z 'Disk-Header                            [0260]
55 POKE &AC3E,&41 'Fehler in Disassemble
   r korrigieren                              [2E0A]
60 P=&811D:GOSUB 400 'Einschalt-Paramete
   rblock                                     [F580]
65 SAVE"SMON",B,&8000,&F5A,&8000              [FDD6]
70 END                                        [B7BC]
100 DATA &5E,&A4,&86,&B2,&07,&B8              [B8BA]
200 DATA &5B,&A8,&27,&B7,&1F,&B1              [B2D0]
300 DATA &5E,&A8,&27,&B7,&1F,&B1              [2DD8]
350 DATA MODE,PAPER,PEN,BORDER                [7D68]
400 RESTORE 350                               [A120]
410 FOR I=0 TO 3                              [B510]
420    READ MSG$:PRINT:PRINT"ALTER WERT F
    UER ";MSG$;":";PEEK(P+I)                  [D49E]
430    PRINT"NEUER WERT FUER ";MSG$;:INPU
    T": ",X:POKE P+I,X                        [F0B2]
440 NEXT                                      [66EE]
450 RETURN                                    [AB32]
```

**Listing. Mit diesem Programm wird der SMON noch besser**

## Light-Cycles läuft ...

... mit einer kleinen Änderung auf allen drei CPC-Modellen. In dem im 3. Schneider-Sonderheft ab Seite 92 abgedruckten Listing müssen Sie lediglich in den drei Zeilen 334, 336 und 337 folgende Anpassung vornehmen: Sie ersetzen den Wert 26 jeweils durch 53 (CPC 664) beziehungsweise 63 (CPC 6128). Hierdurch wird das niederwertige Adreßbyte der Betriebssystem-Routine »MC-Soundregister« an die beiden CPC-Modelle angepaßt.

(Edda Nerz/ma)

## Und noch eine Anpassung

Das Mathematik-Programm aus dem 3. Schneider-Sonderheft (Seite 107) läuft auch auf den beiden Modellen CPC 664 und 6128 einwandfrei, wenn Sie die Hardcopy-Routine ab Programmzeile 1000 gegen die Hardcopy-Routine aus Happy-Computer, Ausgabe 6/86 auf Seite 80 austauschen.

(ma)

## Fehler im Detail

Aufmerksame Leser haben uns auf einen kleinen Fehler im zweiten Schneider-Sonderheft auf Seite 85 (»Daten im direkten Zugriff«) hingewiesen. Dort gibt das Listing 1 die Befehlsfolge zum Speichern des Maschinencode-Programms fälschlicherweise mit einer Länge von 18180 Byte (hex) an. Ein Programm dieser Länge würde den Speicher des CPC natürlich hoffnungslos überfüllen. Deshalb lautet der korrekte Befehl zum Speichern des Programms:

`SAVE"erwbin",b,&A000,&180`

(ma)

## Dem Frust ein Ende

Um Mißverständnissen im Artikel »Datenübertragung muß nicht teuer sein« aus dem 2. Schneider-Sonderheft (Seite 14) vorzubeugen, weisen wir auf Folgendes hin: Die Pin-Numerierung des Bausteins 6850 bezieht sich auf den direkten Anschluß an den Sockel des Z80-Prozessors. Wenn Sie die serielle Schnittstelle V.24 mit dem Erweiterungsanschluß verbinden, müssen Sie die Pins entsprechend ihrer Funktion umnumerieren.

Am Baustein 74LS393 muß es Pin 9 statt Pin 10 heißen. Der 74LS04 arbeitet als Inverter und der 74LS08 übernimmt die Funktion der AND-Gatter.

(ma)

Seite 115 hatten wir zum leichteren Eingeben in Zehnerabstände umnumeriert. Leider wurde dabei vergessen, das Schema zur Änderung der Gruppen mit umzuwandeln.

Die Namen der drei Gruppen stehen in den Programmzeilen 3550 bis 3570. Die Variablen az (Anzahl der Mannschaften) und gruppe$ (Name der Gruppe) sowie die zugehörigen DATA-Zeilen stehen in 4020 bis 4050, 4070 bis 4100 und 4120 bis 4150. Die einzelnen Begegnungen müssen Sie in den Zeilen 6280 bis 6440 angeben.

Auf Seite 122 fehlt im Druckprogramm für Banküberweisungen die letzte Zeile. Sie lautet »2460 RETURN«.

Die Türme von Hanoi auf Seite 136 werden nur ab- und aufgebaut, wenn die vorletzte Zeile im ersten Block (to TvH) »run item:wahl simul selbst« lautet. In der Routine »fertig« fehlt außerdem das abschließende »end«. (ma)

---

# Jetliner – jetzt noch schöner als Fliegen

Der Flugsimulator »Jetliner« aus unserem 3. Schneider-Sonderheft (Happy-Computer-Sonderheft 4/86) faszinierte eine Vielzahl unserer Leser. Das zeigte uns die Menge an Briefen und Telefonaten zu diesem Programm. Einziger Wermutstropfen war die relativ schwierige Bedienung, die vor allem unerfahreneren Piloten das Fliegerleben schwer machte. Jetzt bieten wir allen begeisterten Flugkapitänen eine Verbesserung ihres Jetliners an. Und das Tollste daran: Diese Aufrüstung bedarf keines Werkstattaufenthalts. Die drei zusätzlichen Ausstattungen:

### Übungs-Landeanflug

Wählen Sie im Hauptmenü <X>, versetzt der CPC Ihr Flugzeug sofort zum Beginn des Spiels in die Luft. Die Entfernung zum Flughafen beträgt dabei zwischen 20 und 35 Meilen. Daraus ergeben sich zwei Vorteile. Erstens können Sie nun den sehr schwierigen Landeanflug beliebig oft üben, ohne jedesmal den weiten Anflugweg zurücklegen zu müssen. Zweitens gelangen Anfänger schneller zum verdienten Erfolgserlebnis.

### Spielstand speichern

Bevor Sie bei längeren Flügen riskante Flugmanöver ausprobieren, empfiehlt es sich, in Zukunft den Spielstand durch Druck der Taste <R> unter dem Namen des Piloten zu speichern. Bei Kassettenbetrieb müssen Sie schon vorher die Tasten <REC> und <PLAY> drücken, denn die Speicherung beginnt augenblicklich.

### Spielstand laden

Drücken Sie während des Starts oder des Fluges die Taste <E>, lädt der Computer den gespeicherten Spielstand als neuen Ausgangspunkt.

Um die neuen Funktionen zu nutzen, geben Sie bitte das Listing ein und speichern es als ASCII-Datei (beispielsweise mit »SAVE "JETMERGE",a«). Danach laden Sie die Urfassung des Jetliner und lassen die neuen und geänderten Zeilen mit dem Befehl »MERGE "JETMERGE"« automatisch einfügen. Die nun fertige neue Version speichern Sie schließlich und gehen an den ersten Probeflug. Wenn die Änderungen wie gewünscht funktionieren, benötigen Sie zukünftig weder den alten Jetliner noch das Listing des Patchs, so daß Sie diese später beruhigt löschen dürfen.

(Claus Herwig/ja)

```
90 CLEAR                                     [6CE0]
150 lg=4                                     [66F4]
750 CLS:LOCATE 10,12:INPUT"Name des Pilo
   ten";name$                                [E30C]
1031 LOCATE 25,15:PRINT"Lande;bung....X"
                                             [2838]
1051 IF a$="x" THEN GOSUB 8000               [35D6]
2005 IF ueb=1 THEN GOSUB 8050                [B838]
2113 IF b$="e" THEN 8200                     [FA76]
2115 IF b$="r" THEN 8300                     [1F96]
5745 IF ueb=1 THEN 1300                      [5302]
5760 IF b8>zz+200 THEN l6=zz                 [8FD8]
5761 IF ueb=0 THEN RETURN                    [1B34]
8000 a$=CHR$(INT(RND*10)+48)                 [E77C]
8010 ueb=1                                   [E826]
8020 RETURN                                  [ED94]
8050 a2=15+INT(RND*10)                       [84F2]
8060 a1=15+INT(RND*10)                       [82F2]
8062 hh=INT(RND*100)                         [C4B0]
8065 IF RND(1)>0.5 THEN a1=a1*-1             [9662]
8067 IF RND(1)>0.5 THEN a2=a2*-1             [CB6A]
8068 IF RND(1)>0.5 THEN hh=hh*-1             [F760]
8069 a3=a3+hh:IF a3>=360 THEN a3=a3-360
     ELSE IF a3<0 THEN a3=a3+360             [A364]
8070 b8=45*INT(RND*100)+zz                   [90A6]
8080 d1$="7":d1=7:b7=350:b4=0:b6=1:f3=1:
     l=7:asou=388:n7=15679:ueb=0             [1102]
8120 LOCATE wmot,smot:PEN 1:PRINT STRING
     $(3,CHR$(231))                          [07C2]
8130 RETURN                                  [AA98]
8200 REM laden                               [B8E4]
8205 OPENIN"!jl"+RIGHT$(name$,6)             [2570]
8210 INPUT #9,a1,a2,a3,a4,asou,abk1,abk2
                                             [ECE0]
8215 INPUT #9,b3,b4,b5,b6,b7,b8,bk1,bk2      [8DCC]
8220 INPUT #9,d1,d2,d6,dl,d8                 [A50A]
8225 INPUT #9,e1,eo,e6,e7,ef,er,e4,e5        [94A6]
8230 INPUT #9,f3,f1,f4,f5,fr,fa              [8F0A]
8235 INPUT #9,gr,h2,h1,h3,h4,h5,h6,h7,h8
     ,hr                                     [6A9A]
8240 INPUT #9,k8,k2,k3,k6,k9,ktrl            [87C4]
8245 INPUT #9,l6,l7,l,lb1,lb                 [21B2]
8250 INPUT #9,n7,sq1,t1,zf,ww,ru,ti1,ti2
     ,sd,trv,zz,zx,sou,xc                    [BCBE]
8255 INPUT #9,bn1$,bn$,d4$,d1$               [6F7C]
8260 PEN 1:LOCATE wmot,smot:PRINT STRING
     $(3,CHR$(231)):IF eo=1 THEN PEN 2:L
     OCATE wmot+2,smot:PRINT CHR$(231)       [E952]
8265 LOCATE wff,sff:PEN 1:PRINT bn$          [D688]
8270 LOCATE wfw,sfw:PEN 2:IF k8=1 THEN P
     RINT CHR$(134)CHR$(132) ELSE PRINT
     CHR$(137)CHR$(129)                      [3EC2]
8275 LOCATE wkln,skln:IF k6=0 THEN PEN 2
     :PRINT USING"#";k6 ELSE PEN 1:PRINT
      USING"#";k6                            [170C]
8280 LOCATE wkl,skl:IF k6=0 THEN PEN 2:P
     RINT CHR$(208) ELSE PEN 1:PRINT CHR
     $(204)                                  [D6D0]
8290 CLOSEIN                                 [5A00]
8295 GOTO 2020                               [792A]
8300 REM speichern                           [B160]
8305 OPENOUT"!jl"+RIGHT$(name$,6)            [1B34]
8310 PRINT #9,a1;a2;a3;a4;asou;abk1;abk2
                                             [8D90]
8315 PRINT #9,b3;b4;b5;b6;b7;b8;bk1;bk2      [259A]
8320 PRINT #9,d1;d2;d6;dl;d8                 [4E7E]
8325 PRINT #9,e1;eo;e6;e7;ef;er;e4;e5        [B374]
8330 PRINT #9,f3;f1;f4;f5;fr;fa              [869C]
8335 PRINT #9,gr;h2;h1;h3;h4;h5;h6;h7;h8
     ;hr                                     [B6A4]
8340 PRINT #9,k8;k2;k3;k6;k9;ktrl            [1B56]
8345 PRINT #9,l6;l7;l;lb1;lb                 [1426]
8350 PRINT #9,n7;sq1;t1;zf;ww;ru;ti1;ti2
     ;sd;trv;zz;zx;sou;xc                    [9A40]
8355 PRINT #9,bn1$:PRINT #9,bn$:PRINT #9
     ,d4$:PRINT #9,d1$                       [E50A]
8390 CLOSEOUT                                [5EC4]
8395 GOTO 2020                               [D92C]
```

**Listing. Der beliebte Flugsimulator »Jetliner« noch besser als bisher**

**Große Umfrage**

# Hat Ihnen das Heft gefallen?

Wieder einmal haben Sie ein Schneider-Sonderheft von Happy-Computer vor sich liegen. Und wieder fragen wir uns, ob wir mit unseren Themen richtig liegen. Denn diese Frage können nur Sie – unsere Leser – beantworten. Deshalb schicken Sie uns bitte den untenstehenden Fragebogen ausgefüllt zurück. Denn seine Auswertung zeigt uns den Weg, den wir mit dem 7. Schneider-Sonderheft einschlagen müssen.

Auch der Schneider-Teil im Stamm-Magazin Happy-Computer wird nach Ihren Vorschlägen gestaltet. Deshalb ist Ihre Meinung für uns so immens wichtig.

Und damit sich die Anstrengung für Sie lohnt, verlosen wir unter allen Einsendern eine Reise zur CeBIT 1987 nach Hannover. Informieren Sie sich direkt auf der interessantesten Computermesse Deutschlands.

Schicken Sie den ausgefüllten Fragebogen bis zum 31. Januar 1987 (Datum des Poststempels) an:

**Markt&Technik Verlag AG**
**Redaktion Happy-Computer**
**Kennwort: Schneider-Umfrage**
**Hans-Pinsel-Straße 2**
**8013 Haar bei München**

(Der Rechtsweg ist ausgeschlossen) (hg)

## Fragebogen zum 6. Schneider-Sonderheft

**Wie hat Ihnen dieses Heft gefallen?**

☐ sehr gut
☐ gut
☐ mittel
☐ weniger gut
☐ gar nicht

**Welche Rubriken wollen Sie in Zukunft erweitert sehen?**

☐ Hardware
☐ Software
☐ Basteleien
☐ Spiele-Tests
☐ CP/M
☐ PC-Teil (MS-DOS)
☐ Grundlagen
☐ Einsteiger-Teil
☐ Aktuelles
☐ Tips&Tricks
☐ Spiele-Listings
☐ Anwendungs-Listings
☐ Grafik-Listings

**Welche Rubriken sollen in Zukunft eingeführt werden?**

______________________

______________________

**Welche Computer-Zeitschriften lesen Sie?**

☐ Happy-Computer
☐ deutsche Schneider-Zeitschriften – wenn ja, welche?

______________________

______________________

☐ englische Amstrad-Zeitschriften
☐ andere – wenn ja, welche?

______________________

______________________

**Welche Schneider-Sonderausgaben von Happy-Computer haben Sie sich schon gekauft?**

☐ 1. Schneider-Sonderheft
☐ 2. Schneider-Sonderheft
☐ 3. Schneider-Sonderheft
☐ 4. Schneider-Sonderheft
☐ 5. Schneider-Sonderheft

**Wenn Sie alle sechs Schneider-Sonderhefte besitzen, welches hat Ihnen am besten gefallen?**

☐ 1. Schneider-Sonderheft
☐ 2. Schneider-Sonderheft
☐ 3. Schneider-Sonderheft
☐ 4. Schneider-Sonderheft
☐ 5. Schneider-Sonderheft
☐ 6. Schneider-Sonderheft

**Welchen Computer besitzen Sie?**

☐ Schneider CPC 464
☐ Schneider CPC 664
☐ Schneider CPC 6128
☐ Schneider Joyce
☐ Schneider PC
☐ einen anderen, welchen?

______________________

**Welchen Diskettencontroller besitzen Sie?**

☐ Schneider
☐ Vortex
☐ Vortex X-Controller
☐ einen anderen, welchen?

______________________

**Welche Speichererweiterung besitzen Sie?**

☐ Data Media
☐ dk'tronics
☐ Vortex
☐ eine andere, welche?

______________________

Ich bin damit einverstanden, daß die hier gemachten Angaben elektronisch verarbeitet werden.

Name/Vorname ______________________

Straße ______________________

PLZ/Ort ______________________

Alter ______ Jahre

# Impressum

**Herausgeber:** Carl-Franz von Quadt, Otmar Weber

**Chefredakteur:** Michael Scharfenberger (sc)
**Stellv. Chefredakteur:** Michael Lang (lg)
**Redakteure:** Gregor Neumann (gn), Andreas Hagedorn (hg), Heinrich Lenhardt (hl), Thomas Jacobi (ja), Joachim Graf (jg), Martin Aschoff (ma);
**Chef v. Dienst:** Petra Wängler
**Schlußredaktion:** Eva Hierlmeier
**Redaktionsassistenz:** Monika Lewandowski (222), Rita Gietl (289)

**Fotografie:** Jens Jancke

**Titelgestaltung:** Heinz Rauner Grafik-Design

**Layout:** Leo Eder (Ltg.), Rolf Raß, Katja Milles

**Produktionsleiter:** Klaus Buck (180)

**Anzeigenverkaufsleitung:** Ralph-Peter Rauchfuss

**Auslandsrepräsentation:**
**Schweiz:** Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. (042) 41 56 56, Telex: 862329 mut ch
**USA:** M&T Publishing Inc., 501 Galveston Dr., Redwood City, CA 94063; Tel. 415-366-3600, Telex 752-351

**Manuskripteinsendungen:** Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Anzeigenverkauf:** Britta Fiebig (211), Helmut Distl (398)

**Anzeigenverwaltung und Disposition:** Patricia Schiede (172)

**Marketingleiter:** Hans Hörl (114)

**Vertriebsleiter:** Helmut Grünfeldt (189)

**Vertrieb Handelsauflage:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Str. 96, 7000 Stuttgart 1, Tel. (07 11) 6483-0

**Bezugsmöglichkeiten:** Leser-Service: Telefon (089) 46 13-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

**Bezugspreis:** Das Einzelheft kostet DM 14,–

**Druck:** SOV St. Otto-Verlag GmbH, Laubanger 23, 8600 Bamberg





**Verantwortlich:**
Für redaktionellen Teil: Michael Scharfenberger
Für Anzeigen: Britta Fiebig

**Redaktionsdirektor:** Michael M. Pauly

**Vorstand:** Carl-Franz von Quadt, Otmar Weber

**Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:**
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 46 13-0, Telex 5-22 052

**Telefon-Durchwahl im Verlag:**
**Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen (089) 46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.**


# Markt&Technik DEPOT-BUCHHÄNDLER

**Buchhandlung Herder**, Kurfürstendamm 69
**1000** Berlin 15, Tel. (030) 8835002, BTX *921782#
**Computare Fachbuchhandlung**, Keithstraße 18
**1000** Berlin 30, Tel. (030) 2139021
**Thalia Buchhaus**, Große Bleichen 19
**2000** Hamburg 36, Tel. (040) 3005050
**Boysen + Maasch**, Hermannstraße 31
**2000** Hamburg 1, Tel. (040) 3005050
**Electro-Data**, Wilhelm-Heidsiek-Straße 1
**2190** Cuxhaven, Tel. (04721) 51288
**Buchhandlung Muehlau**, Holtenauer Straße 116
**2300** Kiel, Tel. (0431) 85085
**ECL**, Norderstraße 94-96
**2390** Flensburg, Tel. (0461) 28181
**Buchhandlung Weiland**, Königstraße 79
**2400** Lübeck, Tel. (0451) 160060
**Buchhandlung Storm**, Langenstraße 10
**2800** Bremen 1, Tel. (0421) 321523
**Buchhandlung Lohse-Eissing**, Marktstraße 38
**2940** Wilhelmshaven, Tel. (04421) 41687
**Buchhandlung Schmorl u. v. Seefeld**, Bahnhofstraße 13
**3000** Hannover 1, Tel. (0511) 327651
**Buchhandlung Graff**, Neue Straße 23
**3300** Braunschweig, Tel. (0531) 49271
**Deuerlich'sche Buchhandlung**, Weender Straße 33
**3400** Göttingen, Tel. (0551) 56868
**Buchhandlung an der Hochschule**, Holländische Straße 22
**3500** Kassel, Tel. (0561) 83807
**Stern Verlag**, Friedrichstraße 24-26
**4000** Düsseldorf, Tel. (0211) 373033
**Buchhandlung Baedeker**, Kettwiger Straße 33-35
**4300** Essen 1, Tel. (0201) 221381
**Regensberg'sche Buchhandlung**, Alter Steinweg 1
**4400** Münster, Tel. (0251) 40541-5
**Buchhandlung Acker**, Johannisstraße 51
**4500** Osnabrück, Tel. (0541) 28488
**Buchhandlung C.L.Krüger**, Westenhellweg 9
**4600** Dortmund, Tel. (0221) 1527358
**Buchhandlung Brockmeyer**, Querenburger Höhe 281/Unicenter
**4630** Bochum, Tel. (0234) 701360
**Buchhandlung Meier + Weber**, Warburger Straße 98
**4790** Paderborn, Tel. (05251) 63172
**Buchhandlung Phönix GmbH**, Oberntorwall 25
**4800** Bielefeld 1, Tel. (0521) 58306-38
**Buchhandlung Gonski**, Neumarkt 24
**5000** Köln 1, Tel. (0221) 210528
**Mayer'sche Buchhandlung**, Ursulinerstraße 17-19
**5100** Aachen, Tel. (0241) 4777-136
**Buchhandlung Behrendt**, Am Hof 5a
**5300** Bonn 1, Tel. (0228) 658021
**Buchhandlung Cusanus**, Schloßstraße 12
**5400** Koblenz, Tel. (0261) 36239
**Akad. Buchhandlung Interbook**, Fleischstraße 61-65
**5500** Trier, Tel. (0651) 43596
**Buchhandlung W. Finke**, Kipdorf 32
**5600** Wuppertal 1, Tel. (0202) 454220
**Buchhandlung Balogh**, Sandstraße 1
**5900** Siegen, Tel. (0271) 55298-9
**Buchhandlung Naacher**, Steinweg 3
**6000** Frankfurt 1, Tel. (069) 298050
**Buchhandlung Wellnitz**, Lautenschlägerstraße 4
**6100** Darmstadt, Tel. (06151) 76548
**Buchhandlung Feller + Gecks**, Friedrichstraße 31
**6200** Wiesbaden, Tel. (06121) 304911
**Ferber'sche UNI-Buchhandlung**, Seltersweg 83
**6300** Gießen, Tel. (0641) 12001
**Sozialwissenschaftliche Fachbuchhandlung**, Friedrichstraße 24
**6400** Fulda, Tel. (0661) 75077
**Albertis-Hofbuchhandlung**, Langstraße 47,
**6450** Hanau, Tel. (06181) 24301
**Gutenberg Buchhandlung**, Große Bleiche 29
**6500** Mainz, Tel. (06131) 37011
**Buchhandlung Bock + Seip**, Futterstraße 2
**6600** Saarbrücken, Tel. (0681) 30677
**Buchhandlung Wilhelm Hofmann**, Bismarckstraße 98
**6700** Ludwigshafen, Tel. (0621) 516001
**Buchhandlung Loeffler**, B 1,5
**6800** Mannheim 1, Tel. (0621) 28912
**Buchhandlung Stehn**, Bahnhofstraße 13
**7000** Stuttgart 50, Tel. (0711) 561476
**Osiandersche Buchhandlung**, Sindelfinger Allee 25
**7030** Böblingen
**Buchhandlung am Markt**, Kramstraße 6
**7100** Heilbronn, Tel. (07131) 68682
**UNI Buchhandlung Kellner + Moessner**, Kaiserstraße 18
**7500** Karlsruhe, Tel. (0721) 691436
**Osiandersche Buchhandlung**, Wilhelmstr. 12
**7400** Tübingen, Tel. (07071) 51761
**Osiandersche Buchhandlung**, Kaiserpassage 8
**7410** Reutlingen
**Buchhandlung Roth**, Hauptstraße 45
**7600** Offenburg, Tel. (0781) 22097
**Rombach Center**, Bertoldstraße 10
**7800** Freiburg, Tel. (0761) 49091
**Fachbuchhandlung Hofmann**, Hirschstraße 4
**7900** Ulm, Tel. (0731) 60949
**Schauties Elektronik**, Wangener Str. 99
**7980** Ravensburg, Tel. (0751) 26138
**Buchhandlung Hugendubel**, Marienplatz
**8000** München 2, Tel. (089) 2389-1
**Computerbücher am Obelisk**, Barerstraße 32-34
**8000** München 2, Tel. (089) 282383
**Pele's Computerbücher**, Schillerstraße 17
**8000** München 2, Tel. (089) 555229
**Universitätsbuchhandlung Lachner**, Theresienstraße 43
**8000** München 2, Tel. (089) 521340
**Buchhandlung Schönhuber**, Theresienstraße 6
**8070** Ingolstadt, Tel. (0841) 33146/47
**Computerstudio Gertrud Friedrich**, Ludwigstraße 3
**8220** Traunstein, Tel. (0861) 14767
**Buchhandlung Pustet**, Kl. Exerzierplatz 4
**8390** Passau, Tel. (0851) 56945
**Buchhandlung Pustet**, Gesandtenstraße 6
**8400** Regensburg, Tel. (0941)53061
**Universitätsbuchhandlung Büttner & Co.**, Adlerstraße 10-12
**8500** Nürnberg, Tel. (0911) 2368-0
**Computer-Center-Burger**, Leimitzer Straße 11-13
**8670** Hof, Tel. (09281) 40075
**Sortiments- u. Bahnhofsbuchh. J. Strykowski**, Bahnhofplatz 4
**8700** Würzburg, Tel. (0931) 54389
**Buchhandlung Pustet**, Grottenau 4
**8900** Augsburg, Tel. (0821) 35437
**Kemptener Fachsortiment**, Salzstraße 30
**8960** Kempten, Tel. (0831) 14413

**Schweiz:**
**Buchhandlung Francke AG**, Neuengasse 43, Von-Werdt-Passage
**3001** Bern, Tel. (031) 221717
**Buchhandlung Scherz**, Marktgasse 25
**3011** Bern, Tel. (031) 226837
**Buchhandlung Meissner**, Bahnhofstrasse 41
**5000** Aarau, Tel. (064) 247151
**Bücher Balmer**, Neugasse 12
**6300** Zug, Tel. (042) 214141
**Buchhandlung Enge**, Bleicherweg 56
**8002** Zürich, Tel. (01) 201 2078
**Buchhandlung Orell Füssli**, Pelikanstrasse 10
**8022** Zürich, Tel. (01) 2118011
**Freihofer AG, Wissenschaftliche Buchhandlung**, Universitätsstrasse 11
**8033** Zürich, Tel. (01) 3634282
**Buchhandlung am Rösslitor**, Webergasse 5
**9001** St. Gallen, Tel. (071) 228726

**Österreich:**
**Morawa & Co**, Wollzeile 11
**1010** Wien, Tel. (0222) 947641
**Computer Buch Shop Karl Fegerl**, Heinertstraße 3
**1020** Wien, Tel. (0222) 245368
**Lehrmittelzentrum**, Karlsplatz 13
**1040** Wien, Tel. (0222) 567801
**Johann Reisinger**, Hauptplatz 30, Kirchenstraße 3
**3302** Amstetten, Tel. (07472) 2576-0
**Helmut Lainer**, Obere Landstraße 8
**3500** Krems, Tel. (02732) 2818
**R. Pirngruber**, Landstraße 34
**4020** Linz, Tel. (0732) 272834
**Buchhandlung Schachtner**, Stadtplatz 28
**4840** Vöcklabruck, Tel. (07672) 3467
**R. Regelsberg**, St.-Jullen-Straße 2
**5020** Salzburg, Tel. (0662) 73573
**Tyrolia**, Maria-Theresien-Straße 15
**6010** Innsbruck, Tel. (05222) 24944
**Wagner'sche Universitätsbuchhandlung**, Museumstraße 4
**6010** Innsbruck, Tel. (05222) 22316
**Buchhandlung Leykam**, Stemplergasse 3
**8010** Graz, Tel. (0316) 76676-0
**Jos. A. Kienreich**, Sacherstraße 6
**8010** Graz, Tel. (0316) 76441
**Volksbuchhandlung**, Radetzkystraße 7
**8010** Graz, Tel. (0316) 79388




Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München





DIE ERBSCHAFT
PANIK IN LAS VEGAS

# Alles Rouletti?

Die Aufgabe ist simpel: Aus 200 Dollar innerhalb einer Nacht in Las Vegas eine Million machen, sonst wird es nichts mit der riesengroßen Erbschaft der alten Tante. Der Haken dabei: Du sitzt 4000 Kilometer von Las Vegas entfernt, hast nur 200 Dollar, das Flugticket, und vor der Tür lauert ein Haufen Gläubiger, die sofort ihr Geld haben wollen. Alles rouletti?



Wer wissen will, was wir außer der Erbschaft noch so alles haben, dem schicken wir gerne unseren Gesamtkatalog zu.

Name ______________________

Straße ______________________

PLZ __________ Ort __________

An: ariolasoft, Carl-Bertelsmann-Str. 161, 4830 Gütersloh.




Happy SH 1/87

SPITZE SACH ICH!

Nee, ja, wieso? Kennste nich?
Werner in disk! Was Du brauchst?
Jaaa, Nervenkostüm, Hang zum Glücksspiel, Freunde (guute), Würfelbecher + natürlich, Kamillentee, Flens, Honich, kleines Moderratt (zum Simulieren), zuverlässigen Verkehrsfunk (Werner sacht, gib's nich), nee, ja un dann geht's los!

Lot Di man ni griepen!! Sacht Werner. Kommt dann der Gesamtkatalog. Aber Hallo!

Name ______________________

Straße ______________________

PLZ __________ Ort ______________

An: ariolasoft, Carl-Bertelsmann-Str. 161, 4830 Gütersloh.




SSS 2.87